

Commodore INFO

PRIJS ƒ 7,95 / Bfr. 160

ONAFHANKELIJK BLAD VOOR AMIGA- EN C64- GEBRUIKERS Jaargang 7, NO.5, juli/augustus '90

- De Amiga familie
- Fantavision
- CES
- Deluxe Paint III
- Van 64 naar PC
- Deluxe Video 3

Inclusief 25 pagina's Amiga nieuws

COLOFON

Commodore INFO is een uitgave van:
Sala Communications
Postbus 43048
1009 ZA Amsterdam

Uitgever:
V.Sharfman

Redactie:
ir. L. Sala (hoofdredacteur),
W.A. Scheer (eindredacteur),
drs. U. Schuurmans,
R.Goudriaan, B. Venema,
P. Boncz, MGCC/Johan&Johan,
Michel de Boer, Hylke Sprangers,
Rosa van Tijn (secretaresse)
telefoon: 020-228871

Produktie:
drs. H. Zoete,
J. Broekhuizen

Advertentie-exploitatie:
ing. V. Sala, D. van Vlijmen
ing. B. Sala, H. Bia
telefoon: 020-273198

Abonnementen en administratie:
Marjo Jansen
telefoon: 020-248006

Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket. **Telefonisch uitsluitend tussen 12.00 en 15.00 uur.**

Vragen over artikelen:
Voor vragen en opmerkingen over artikelen of anderszins graag een briefje aan de redactie (zie boven).

Listingtelefoon:
02155-25162
(ma: 17.00-21.00 u)
(In juli en augustus niet)

Illustraties:
Ben van Mierlo

Cover:

Imagesetting IFF:
3Gitaal, Amsterdam

Zetwerk & druk:
NDB, Zoeterwoude

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud



Redactioneel

Commodore leeft. Ik weet niet of de zomer invloed heeft, maar de programma's, updates en zelfs nieuwe machines (bijvoorbeeld de Amiga 3000) lijken wel als bollen uit de grond te schieten.

In deze Commodore Info is er dan ook weer volop nieuws. Naast de gebruikelijke aandacht voor de C-64 en de C-128 ook weer ruim aandacht voor de Amiga. Naast veel nieuwe software, een uitgebreide bespreking van de Amiga 3000, hebben we de hele Amiga range van produkten nog eens onder de loep genomen.

Verder vernamen we net voor het verschijnen van dit nummer dat binnenkort de echte Atari-emulator komt, die qua snelheid niet onderdoet voor de Atari zelf. Deze vervangt de pre-release versie die al enige tijd in omloop is, maar niet echt goed functioneerde. Naast de Apple en Dos emulatoren is de Amiga hiermee een van de meest complete computers geworden, zeker wanneer we de uitgebreide mogelijkheden van de Amiga 3000 in ogenschouw nemen. Kortom de Amiga bezitters kunnen nog jaren vooruit met hun machines en behoeven zich geen zorgen te maken dat er te weinig software -support zou zijn.

Wilbert Scheer

Listings C-64

Abonnement:

Tot het einde van het jaar (4 nummers): *f* 30,- of Brf. 510
Betaling op giro 4985259 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info.
Oude nummers kunt u alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening.
Ook telefonisch bestellen is mogelijk. Bel GRATIS 06-02242222 (teleservice) elke dag van 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze nummers zijn allen bedoeld voor de opgave van nieuwe abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

CES CHICAGO

Consumenten electronica in de VS in het dal, CD-ROM voor Amiga aangekondigd.

De Consumer Electronics Show in juni in Chicago is de graadmeter voor wat Amerika op audio-en video gebied en consumenten- electronica in het hoogseizoen (november/december) te wachten staat. Hier worden de bestellingen geplaatst voor wat rond de Kerst in de winkels ligt. Het is een soort super-Firato, maar dan helemaal gericht op de dealers en electronica-winkeliers. Die moeten nu de orders plaatsen voor wat ze in het najaar denken te kunnen verkopen. Dat lijkt een groots spektakel, maar na ettelijke CES shows hebben we geleerd door de uiterlijke schijn heen te kijken en dan wordt zo'n CES gauw een routinezaak met veel glans en glitter, maar waar verder gewoon zaken gedaan moeten worden.

Matig jaar

Na een vrij slecht eerste halfjaar verwacht men de tweede helft van 1990 wel wat betere verkopen en over het hele jaar zou de CE business in de VS nog wel met 5% kunnen groeien, tot ongeveer 34 miljard dollar dit jaar, zeggen de optimisten. Na de CD golf is echter de kooplust van het publiek wat afgenomen, ook bedrijven als Philips hebben daar wereldwijd last van. De concurrentie neemt toe en de winsten zakken in. De hoop is gevestigd op de verkopen van Camcorders, de grotere TV's, home automation en personal fax. Personal computers voor thuis en het veel genoemde 'thuiskantoor' zijn een beetje in de luwte gekomen.

De opwinding rond de impact van de digitale cassette of DAT voor audio is wat geluwd, nu de prijzen voor dergelijke systemen bekend zijn geworden. Later dit jaar staan ze in de winkel, maar met een prijskaartje van 1000 dollar zijn ze te duur om een massapubliek te bereiken. Maar JVC, Sony, Onkyo en Matsushita gaan nu dus, na veel geharrewar over kopieerproblemen, echte DAT systemen leveren, die wel zijn uitgerust met de door Philips ontwikkelde Serial Copy Management Systemen (SCMS). Het probleem met de DAT is dat er nog nauwelijks voorbespeelde tapes met muziek zijn uitgebracht, de software industrie blijft achter.

Morita

De belangrijkste spreker op de CES was Akito Morita, de baas van Sony. Hij ging in op de integratie van de hardware en software, waar Sony nu ook mee bezig is door b.v. filmbedrijven en studio's te kopen. De makers van apparatuur moeten meer samenwerken met filmmakers, artiesten en musici. Hij zag het samengaan van de video- en optische tenchnieken in met name de Laser Disc (LD) als



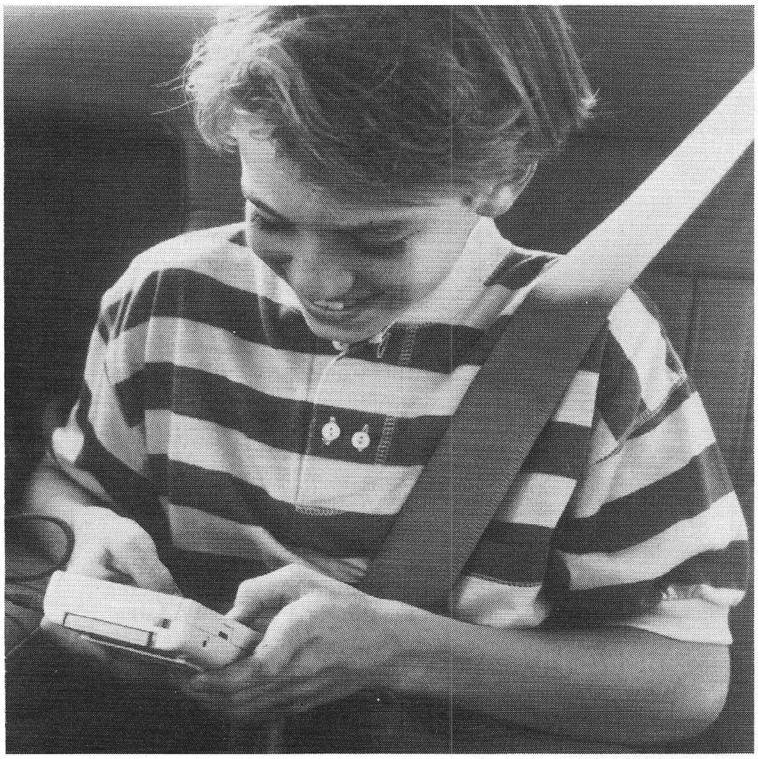
een nieuwe kans voor de makers van hardware en software om samen te werken.

Nadat de VCR de film in de huiskamer bracht, zal de Laser Disc het theater thuis brengen. De veel hogere kwaliteit en extra functies van de LD kan gezien worden als een bedreiging van het 'live' optreden, maar zal naar zijn mening juist stimulerend gaan werken. LD kan gebruikt gaan worden als ondersteuning van live optredens, maar aan de andere kant kunnen met multi-LD spelers compleet nieuwe effecten bereikt worden. Hij ziet voor de toekomst meer in High Definition Video dan in HDTV, dat te beperkt zou zijn. Voor thuis wil men ook de optimale kwaliteit kunnen afspelen, naast wat er via de ether of kabel binnenkomt.

Thuiskantoor

De bezitters van een eigen kantoortje thuis, met eigen PC, fax en copier, dat wordt al jaren als de grote groeimarkt afgeschilderd. Ondanks indrukwekkende marktstatistieken, er zou on-

geveer 17 miljard dollar aan omzet naar toe gaan, valt het allemaal wat tegen. Voor een deel blijkt men thuis niet veel verder te willen gaan dan zogenaamde Personal Word Processors PWP's, waarvan Brother nu een hele reeks heeft.
Een categorie, waarvan men wel succes verwacht zijn de TAMFAXen, een combinatie van personal fax en telefoon beantwoorder. Daarvoor zijn geen twee lijnen nodig, er wordt automatisch omgeschakeld en dergelijke ma-



chines hoeven niet zo duur te zijn. de FAXACE 7 van Mita kost ongeveer 1200 dollar. Nu de professionele faxen steeds meer overgaan op gewoon papier, probeert men de grote produktiecapaciteit in thermische afdrukkers zo naar de thuismarkt om te buigen.

Commodore

Helaas is het zo, dat Commodore tegenwoordig vaker bestuurswisselingen aankondigt dan revolutionaire nieuwe produkten. Iedere keer weer stapels beschrijvingen van nieuwe mensen, die nooit echt indruk maken. Irvin Gould, de grootste aandeelhouder en chairman, zien we nu zelfs van arren moede maar optreden als executive product manager voor sommige nieuwe spullen, schijnbaar zijn er verder geen managers die iets van de produkten begrijpen. Steeds maar nieuwe topmensen, niemand houdt het langer dan een jaar of anderhalf uit. De huidige top komt voornamelijk van Apple vandaan. Natuurlijk vormt West Duitsland, net als voor Atari, nog steeds de belangrijkste markt voor CBM, met name voor PC's. Maar ook daar rommelt het in de top, terwijl men nu ook voor de PC's hier en daar is afgestapt van eigen fabrikaat en Taiwanese machines van o.a. Mitac onder eigen label brengt (bijvb. het SX model).

Amiga CD

We hebben er lang op moeten wachten, maar op de CES kondigde Commodore eindelijk een CD-ROM voor de Amiga aan. Het gaat nog niet om een geintegreerde CD-Amiga, voorlopig is er niet meer aangekondigd dan een losse CD-ROM speler die kan worden aangesloten op de Amiga. Op zich niet schokkend en de combinatie Amiga 500 en CD-ROM, die op de vorige CES aan ingewijden werd getoond, is het dus nog niet. Het valt een beetje tegen dat Commodore de laatste tijd zo traag is met dit soort produktaankondigingen. Het is mooi dat er nu een superdure Amiga 3000 is, maar dat valt toch een beetje buiten het breik van de grote massa Amiga gebruikers. Maar goed, er is nu een CD-ROM drive voor de Amiga!

Belangrijker is, dat men nu Nolan Bushnell, creatief genie en de oprichter van Atari, binnen Commodore heeft gehaald. Hij is, volgens Jack Tramiel wel 'een van de mensen die het snelst geld kunnen opmaken in deze industrie-tak', maar heeft ook heel wat innovaties op zijn naam staan. Pong, de combinatie van Pizza en video-arcade, de sprekende Ruxpin poppen, Nolan is ongetwijfeld een der groten in de consumenten-electronica.

CD-ROM

De meeste CD-ROM toepassingen zijn nog heel serieus, maar wil dit medium echt doorbreken, dan moet het grote publiek er ook iets mee kunnen. Misschien ironisch, misschien toevallig, maar de jongste Philips dochter Head-Start, met een duidelijk Amerikaanse achtergrond uit de Vendex school, liep hiermee voorop. Als eerste heeft men een consumentenmachine gemaakt met zo'n beeldplaatspeler, waardoor toepassingen met een overgrote datahonger nu ook voor thuisgebruik haalbaar zijn.

Markt-verkenning

In zekere zin doet men met deze eerste generatie CD-ROM, zoals we die bij Commodore, Atari en o.a. Head-Start zien een soort testmarketing. Nadat iedereen jarenlang heeft lopen roepen dat CD-ROM het nieuwe publieksmedium zou worden, waagde toch niemand zich op dit gladde ijs. De CD-ROM drive werd een professioneel speeltje voor relatief elitaire toepassingen, om bijvoorbeeld fonts op te zetten of groet databibliotheken voor trainingen. De stap naar het publiek werd niet echt gemaakt, al riep Bill Gates op zijn jaarlijkse CD-ROM meetings daar wel steeds toe op. Japan had de software niet, de Amerikanen zaten in de val doordat de uitgevers het medium niet zagen zitten, de massamarkt moest de redding maar brengen. Ook Audio
Een van de aantrekkelijke dingen van de nieuwste CD-ROM spelers is dat ze ook gewone (audio)-CD's kunnen afspelen. Wie echt hifi kwaliteit wil horen moet de zaak wel doorkoppelen naar een versterker, maar in principe heeft men met de computer nu ook een CD audiobron erbij. Via een adapter voor CD Singles en een stereo koptelefoon kan men nu dus muziek uit de computer beluisteren.

RAM of harde schijf

Een van de eisen bij een CD-ROM drive is wel, dat er daarnaast voldoende tijdelijke opslag in de vorm van RAM of harde schijf is om met de soms grote bestanden goed te kunnen werken. De CD-ROM is te langzaam, en dus is er een tussenstapje nodig. Plaatjes vragen soms erg veel KB's.
Dit is waarschijnlijk ook de reden, dat men de combinatie Amiga 500 en CD-ROM drive waar sprake van was, nog niet heeft uitgebracht. Met een losse CD-ROM drive kan men eerst de gebruikers van

Amiga 100, 2000 en 3000's bedienen. Die hebben vaak een harde schijf en in ieder geval meer geheugen.

Software

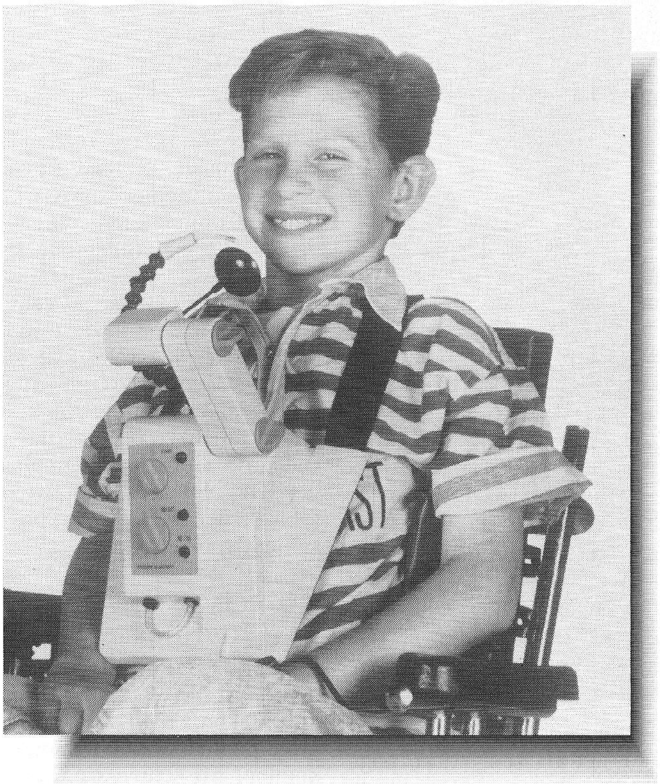
Er is al heel wat spul op CD-ROM voorhanden, al moet de grote 'boom' nog komen. Op dit moment moet ons nog van het hart, dat het CD-ROM aanbod nog wel erg Amerikaans georienteerd is, veel is nog in het engels of slaat op te typisch amerikaanse toestanden. De meest voor de hand liggende toepassing van een CD-ROM zijn natuurlijk de encyclopedische naslag(draai)schijven.
Veel CD-ROM materiaal heeft op een of andere manier een leerdoel, al is het vaak spelend leren. Zo staat de gehele aardbol op de PC-Globe CD-ROM, met uitgebreide zoekfuncties en achtergrond informatie. Verder is er Microsoft Bookshelf, met standaard engelse synoniemen, een almanak etc., de MS Small Business Consultant met nuttige info voor USA kleine ondernemers.

CBM-Promoties

Omdat de budget-PC's ook in de VS steeds goedkoper zijn geworden, moet Commodore steeds allerlei promotie-acties bedenken om haar Amiga aan de man te brengen. De nieuwe Amiga 3000 is wel iets voor de echte fans en professionele video-artiesten, voor de junior-gebruiker ligt de 500 meer binnen de budgetgrenzen. Met een actieve promotiepolitiek en distributie via de zogenaamde mass-market kanalen en warenhuizen gaat men nu de Amiga 500 breed brengen. De 500C wordt in verschillende combinatiepakketten aangeboden, met muis en software zoals Tetris, voor prijzen vanaf 800 dollar. In Europa is de 500 al wat langer op de markt, in de VS begint men nu pas serieus met een brede actie om de markt van jeugdige computer-hobbyisten te bereiken.

Express

Ook heeft Commodore de service-organisatie in de VS een nieuwe injectie gegeven. Met CommodoreExpress biedt men nu een 24 uurs helpline en een door-to-door klanten service voor reparaties.
Ieder jaar vormt de CES het toneel voor de nieuwe computerspelletjes. Dat men daarvoor deze show kiest en niet de professionele computershows als Comdex komt omdat de verkoop van computergames vaak via andere distributiekanalen verloopt.
Een leuke lijkt ons Stratego, ten tegen 'aliens' uit de ruimte, om uiteindelijk de Evil Emperor te ontmoeten. Dit spel komt beschikbaar voor C-64, ST, Amiga, PC.
Klax is een andere hit uit de arcade wereld, die Tengen nu ook voor computers brengt. Dit is een behendigheidsspel in de klassieke video-games traditie, waar snelheid en precisie essentieel zijn. Men moet stenen van een lopende band opvangen en wegzetten, eenvoudig maar fascinerend, iets lijkend op Tetris.
De milieu-trend komt ook in computergames naar voren. Accolade brengt 'Balance of



De NES Hands Free controller is ontwikkeld voor een 12 jarig gehandicapt kind uit Oklahoma.

een van die spellen die tot nu toe ontsnapte aan de computerconversie, uitgebracht door Accolade.
Vindicators van Tengen (Atari games) was al een Nintendo topper, en is nu ook voor andere platforms uitgebracht. Het is een tankslag in de 21ste eeuw, waarbij een of twee spelers hun tank inzetthe Planet', een simulatie van Chris Crawford.
De speler moet als hoge VN commissaris voor het milieu, de wereld zien te redden van milieurampen. Oorzaken en gevolgen van industrie-ontwikkeling, atomaire ongelukken en in totaal 150 van dergelijke milieu-invloeden worden in de simulatie betrokken.
gelukkig zien we dat men nog steeds C-64 versies van dit soort software uitbrengt. Echte 128 spellen zijn er niet meer, ook qua hardware is die machine min of meer geruisloos verdwenen, naar verluidt omdat de produktiekosten in vergelijking met de 64 te hoog waren.

Handheld info

Het electronische boek begint steeds duidelijker te worden. Met de huidige techniek kan men voor een redelijke prijs al heel wat KB's aan ROM in een handheld met een LCD stoppen. Door het comprimeren van de teksten kan bijvoorbeeld 250 KB tekst wel op een 128 KB ROM ruimte.
De displaykwaliteit is nog niet voldoende om het lezen van romans en studieboeken echt plezierig te maken, maar voor meer interactieve boeken werkt het al redelijk. Electronische dicionaires, naslagwerken, spellingscorrectors en vertaalcomputers zijn voorbeelden van dit soort toepassingen. Franklin Electronic Publishers heeft zich gespecialiseerd in dergelijke handhelds. Een complete Columbia Encyclopedia kost in de VS nu 299 dollar. Ook zijn er nogal wat vertaalhandhelds, zoals de Hexaglot Talking translator van Fidelity, die ook Japans aan kan.
De user-interface, hoe haal je de gewenste informatie zo snel mogelijk uit de electronische opslag, is het belangrijkste punt bij de ontwikkeling van dit soort produkten. Wanneer er een goede ingang gevonden is, of gewerkt kan worden met bekende indexen, dan is het maken van een handzaam electronische boek niet zo'n probleem. Selec-Tronics heeft nu ook van de bijbel zo'n electronische uitvoering gemaakt.

Handheld Computer

Flexibeler dan de electronische boeken zijn de handheld

computers, qua functies al heel ver ontwikkeld van de calculators van tien jaar geleden, en vergelijkbaar met de grotere portable computers. Atari's Portfolio begin in de VS in de grote winkelketens op te duiken, maar er is ook stevige concurrentie. Atari heeft nieuwe ROM modules uitgebracht, waarmee de Portfolio weer wat meer te bieden heeft, zoals een met financiële calculaties en een met DOS utilities en een communicatie- programma.
Sharp was een van de eersten op deze markt met de Wizard. Er is nu een nieuwe versie van uit, de Signature. Die is gericht op de zakelijke gebruiker en biedt meer specifieke functies zoals een volledig qwerty toetsenbord, iets groter display (40x8) en meer RAM.
Het wordt druk op de dit gebied, ook andere bedrijven komen met handhelds. Bondwell heeft een notebook-organiser uitgebracht. Met een gewicht van 2,2 kg en wat grotere LCD display dan de Portfolio is het meer een kleine laptop met een V30 CPU en 2 MB RAM en veel ROM software en een ingebouwd 2400 bps modem. De prijs is 999 dollar, maar er is ook een goedkopere versie zonder modem en met 512 KB.
De Casio BOSS is weer wat kleiner en met 64 KB RAM is het meer een super-organiser.



Videogames

De markt voor videogames begint over zijn tweede herlevingspiek heen te raken. In het begin van de jaren tachtig was het Atari, dat hier de huiskamers van de wereld veroverde, de afgelopen jaren profiteerde vooral Nintendo van een grote opleving. De Japanse en Amerikaanse kinderen tussen 10 en 14 jaar groeien op met de Famicon, die in de VS wat formeler Nintendo Entertainment Systeem heet. Maar veel van de 'hits' in deze markt, zoals Teenage Mutant Ninja Turtles van Gametek zijn helemaal niet zo indrukwekkend, en drijven op gillende promotie en de verhuur van modules in videotheken in de VS. Dezelfde ontwerpers, veel van dezelfde spelletjes en meer van hetzelfde kenmerkt deze markt. Zo komt men nu met geluidseffecten en gedigitaliseerde spraak voor de Nintendo, iets wat bij de C-64 al jaren bekend is. Veelal holt men achter de filmsuccesen aan, en 'Batman' en 'Gremlins 2' waren dan ook de toppers op de afgelopen CES.
Niet iedereen doet het goed op deze markt, zoals blijkt uit de prijsverlaging van Sega's 8 bit systeem naar 60 dollar.
Nu het succes wat begint in te zakken probeert men zowel naar de volwassenen als naar de kleinere kinderen toe het aanbod uit te breiden. Voor volwassenen wat gepeperder spelletjes en beeldvarianten van bordspelen als Pictionary, voor kinderen de educatieve richting. 'Non-violent' spelen zijn de nieuwe trend, inspelend op wat ouders willen voor hun kinderen. Harlem Globetrotters is een nieuwe basketball spel, waar men met maximaal vier spelers mee aan de gang kan. Fisher Price gaat nu haar naamsbekendheid en positie op de speelgoedmarkt en de daar gebruikte speelgoedkarakters gebruiken als basis voor spelletjes voor 3 tot 6 jarigen op Nintendo formaat, maar ook voor C-64, Apple en PC.

Naast Nintendo zijn Atari, Sega en Nec actief op deze markt, met 16 bits videospelunits, die feitelijk betere kwaliteit video bieden dan de Nintendo.
Ook de handheld games met verwisselbare spelmodules zoals Atari's Lynx volgen de filmtrends en voor de kleine Nintendo Game Boy portable video games zijn er dan ook al titels als Spiderman en Beetlejuice. Atari heeft nu voor de kleuren-Lynx een spel, Slime World, waarmee acht spelers met aan elkaar gekoppelde Lynxen tegelijk kunnen spelen.

Hands-free

Een interessante ontwikkeling is dat Nintendo nu ook een joystick-achtig besturingsapparaat heeft, dat zonder handen bediend kan worden. Speciaal ontworpen voor gehandicapten werkt het met een zuig/blaas pijpje, de benodigde kracht is instelbaar. Een goede zaak, omdat men hiermee moderne electronica ook bruikbaar maakt voor gebruikersgroepen, die normaal zijn buitengesloten van het computergebeuren.
In dezelfde geest, maar dan voor overactieve gebruikers is er de Power Pad, een mat met aanrakingsgevoelige sensoren op op rond te springen, waarmee men ook videogames kan aansturen.
Interessant voor een groter publiek zijn de 3-D stereoscopische animaties als ORB-IT van Hi Tech Expressions voor de Nintendo. Met een speciale bril kan men dan echt stereo zien en op avontuur gaan in spellen gebaseerd op o.m. Tolkien's Ban van de Ring. Dit wordt de nieuwe generatie electronisch amusement, de zogenaamde 'Virtual Reality'. Op een Nintendo is de kwaliteit nog vrij matig, maar ooit zullen we op deze manier 'virtuele reizen' kunen maken. Ook voor de Atari ST is er al zo'n stereo-bril op de markt.

Video

Het 8 mm formaat voor videocamera's is helemaal terug, Sony heeft hiermee uiteindelijk wel de juiste weg ingeslagen. Ook Hitachi en Samsung gaan nu 8 mm Camcorders maken.
Videoapparatuur wordt steeds kleiner en er wordt om een steeds betere opnamekwaliteit gevraagd, waar Sony met professional 8 mm ook op inspeelt. De Camcorder worden kleiner en beter, en Canon liet een prototype zien, waarbij het optische deel gebruik maakt van standaard 35 MM (SLR fotocamera-maat) uitwisselbare lenzen. De interface daarvoor, VL, wordt gestandaardiseerd. Met name voor natuurfilmers zal dit de mogelijheden van video enorm uitbreiden.

Laser Disc

Nu de CD zo'n brede acceptatie heeft gekregen, ziet de industrie weer wat meer in de optische beeldplaat. Die is een paar jaar geleden, toen in analoge vorm, al eens uitgebracht door o.m. Philips, maar is in die vorm vrijwel vergeten.
Het CD formaat heeft een nieuwe horizon geopend voor optische video zoals het CD/Video formaat bewijst. Door het bouwen van afspeel-units, die zowel de grotere laser-discs (8 en 12 inch) als het CD formaat (CDaudio en CD/Video en 3 inch CD singles) kunnen draaien, is er een opleving in zicht. De veel

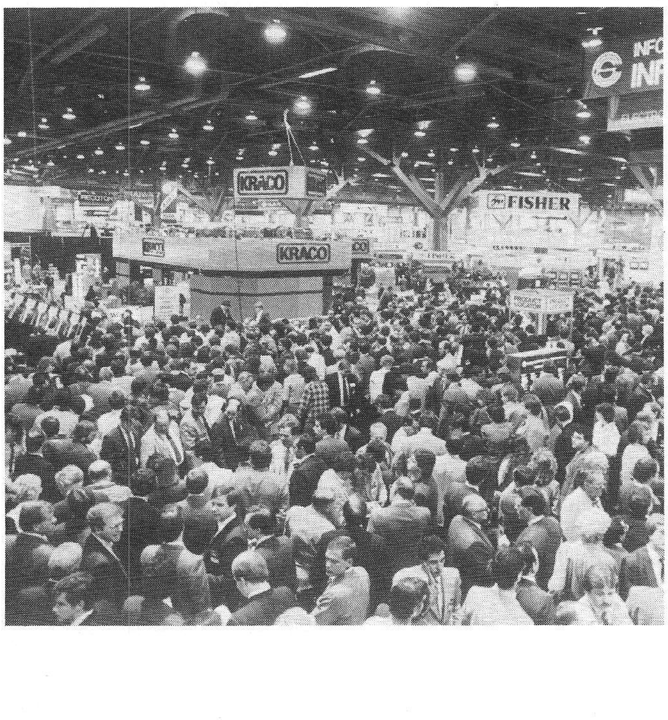
betere weergavekwaliteit vergeleken met video-tape (425 beeldlijnen, dus 40% scherper dan VHS) en zeker de hi-fi audiokwaliteit maken het bekijken van een Laser Disc of CD/V programma duidelijk verschillend van dat van een videotape. Ook zijn de mogelijkheden om heel snel te wisselen en door de beeldinformatie te schieten veel groter, een schijf is veel meer parallel toegankelijk dan een tape.
Er zijn in de VS nu multi-Laser Disc spelers op de markt, zoals de Panasonic LX 200, die rond de 850 dollar kosten. Daarmee wil men klanten bereiken, die eigenlijk op zoek waren naar een CD speler, maar nu alle video-mogelijkheden er bij nemen.
Automatische cameraman
Visionary uit Cambridge Mass. liet op de CES een automatische camera-richtingzoeker zien. Met 'In the Picture' zoekt de camera op een statief met richtingzoeker zelf zijn object en volgt dat ook bij bewegingen. Voor conferenties en opnamen van de familie natuurlijk een uitkomst, er is geen cameraman meer nodig.

Still Video naar computer

De stap van videobeeld naar reproduktie wordt steeds eenvoudiger. Voor de Amiga zijn er al heel wat framegrabber alternatieven, voor de PC en de Mac is dat nog relatief nieuw. Canon heeft samen met Aldus een beeldcapture systeem ontwikkeld voor aansluiting via SCSI op de Macintosh, bedoeld voor professioneel grafisch gebruik, zoals het maken van color-separaties. Kleurenbeeldjes worden met een still-video camera (zoals de RC-470 van Canon) op 2 inch video floppy disks gezet. De kwaliteit van de plaatjes kan gaan tot 640x480 pixels bij 24 bits per pixel. Die worden dan afgespeeld in de FV540 Video Floppy Disk Drive, waar ook de digitalisering wordt verzorgd. Die plaatjes kunnen dan worden geimporteerd in Pagemaker documenten.
De software omvat Pagemaker en Preprint van Aldus en een driver voor de FV540 diskdrive van Canon. Uitdraaien van de documenten kan bijvoorbeeld op de Canon 500 Laser Copier met een speciale PostScript interpreter kaart in 400 dpi kwaliteit.

Home Automation

De laatste jaren probeert men steeds maar weer om de home- automation tot de nieuwe rage te maken. Dat lukt maar matig, veel verder dan wat demonstratie-projecten zoals we ook ins land kennen, komt het eigenlijk niet. De echte produkten voor de massamarkt blijven uit, de Domotica of home-automation slaat nog niet aan. Het lukt niet, om de grote bedrijven echt op dit spoor te krijgen en zonder dat giganten als Philips of Sony in al hun apparatuur de nodige interface-chips gaan bouwen, blijft het bij aan elkaar geknoopte lichtschakelaars en uitgebreide remote/afstandbediening. Dat de grote doorbraak uitblijft wordt geweten aan de steeds maar uitblijvende standaardisatie. Binnen Eureka verband is Philips nu projectleider om tot een Europese standaard voor een homebus te komen, maar men schiet niet erg op. Het lukt ook vrijwel nooit om via een soort industrie-afspraak tot een algemeen geaccepteerde standaard te komen. Veel vaker is er één bedrijf dat het voortouw neemt en een defacto standaard maakt. Voor home automation wil men het dus eens anders doen, ook in de VS werkt men aan een aantal standaards, waaronder die van de Smart House groep en de CEBus van de EIA/CEG Electronics Industries Association (USA 202-4574919). De CEBUS communicatie kan gaan via het electriciteitsnet, via infrarood, coaxiaalkabel, of radiocommunicatie (RF) en in de toekomst via glasfiber kabels. Het moet de electronische ruggegraat van het huis van de toekomst gaan vormen, met functies voor comfort, beveiliging, klimaatregeling en alarmering. Een koppeling naar de telefoon is natuurlijk voorzien, maar de CEBus aanpak is niet echt compatibel met de ISDN aanpak of met de Japanse Home Bus of Europese (Philips) D2B communicatiestandaard. Die CEBus standaard is nu in de evaluatiefase, maar echte produkten volgens die standaard zijn nog niet in zicht. Net als in Europa verschuilt de industrie zich achter mooie voorspellingen en veel commissies en rapporten, waar wordt er niet echt geld tegenaan gegooid.



Custom installation

Wel komen er steeds meer produkten, die audio- en video op meerdere plaatsen in het huis brengen. Gemini levert een soort huis-TV zender, waarmee men het kabelsignaal, maar ook signalen van VCR of bewakingscamera draadloos naar TV's in alle kamers kan doorzenden. (201-4719050). Recotron levert een soortgelijk apparaat, dat Room Service heet.
Om de apparatuur ook vanuit diverse kamers te kunnen bedienen, heeft Videolink een multiroom remote control systeem, dat Hidden Link heet. Met repeaters in iedere kamer kan men zo de centrale installatie van overal bedienen. Ook voor audio- systemen wordt dit steeds meer aangeboden zoals de infrarood retransmitter LeapFrog van Terk Technologies.
Sony heeft een eigen multiroom systeem, dat werkt met coaxiaal kabels terwijl Pioneer werkt met twisted pairs voor de signaal overdracht naar 'slave' units.
Met dit soort installaties, waarvoor veel geld wordt uitgegeven, is er ook een heel nieuwe soort audio/video service gekomen, namelijk de 'Custom AV installers'. Dat zijn bedrijven die voor rijke klanten of bedrijven een compleet audio-video systeem specificeren, leveren en installeren. Dat gaat verder dan hier en daar een box plaatsen, iedereen die wel eens gewerkt heeft met de kabeltroep van moderne installaties ziet in, dat dit al snel specialistenwerk wordt.

I.S.

GEOS-INFO

Vragen te over, zo blijkt bij iedere nieuwe uitgave van Commodore Info. Peter Boncz heeft ook voor dit nummer weer de nodige post gekregen...

Dhr. Jansen uit Tilburg schrijft dat hij onlangs het SonderHeft 48 van het Duitse blad 64-er heeft gekocht. Hierin stond een programma om GEOS uit te breiden, zodat er met 40 tracks per diskette gewerkt kan worden in plaats van de normale 35. Dit levert 21K aan extra ruimte op disk op. Het blijkt echter dat als het programma, 'Plus 21K' genaamd, opgestart wordt, de vraag "Bitte Diskette einlegen mit DeskTop GE 2.0" komt. Er moet dan een disk met de Duitse (GErman) DeskTop-versie in de drive gestopt worden. In Nederland werken we doorgaans met de DeskTop AM (de Amerikaanse versie). Door het flitsende beleid van V&D en de MicroComputer-Club (al geruime tijd ter ziele, dacht ik) is het maken van een Nederlandse GEOS nooit van de grond gekomen. Maar dat terzijde.

Het probleem voor dhr. Jansen blijft dat het "Plus 21 K"-programma een Amerikaanse DeskTop niet accepteert. Wat het programma doet is het vervangen of aanpassen van de Turbodisk-routine, zodat er 40 tracks gelezen en geschreven kunnen worden. Ik begrijp niet goed waar daar de DeskTop voor nodig is (het is dan logischer de GEOS Kernal aan te passen), maar ik leid daaruit af, dat het programma "Plus 21K" de code van de DeskTop wil aanpassen. Maar omdat de in het programma opgeslagen commando-woorden (voor bijvoorbeeld DeskTop-menu's) in het Duits vast niet precies even lang zijn als in het Engels, zal er ongetwijfeld verschil in lengte tussen de DeskTop GE en DeskTop AM optreden. Daarom is het vrijwel zeker dat "Plus 21K" nooit zal werken voor een DeskTop AM. Het is daarom zaak voor dhr. Jansen om aan een Duitse GEOS te komen.

GEOS-club

De volgende vragensteller is misschien wel de andere helft van het beroemde duo. Dhr. Janssen uit Venlo schreef GEOS-INFO een briefje waarin hij informeerde of er misschien een GEOS-club bestond. Welnu, dit is inderdaad het geval. De club heet de Geos Gebruikers Groep en het adres is: Postbus 52, 1330 AB in Almere. De GGG houdt zich bezig met zaken als het verstrekken van informatie over actualiteiten en ontwikkelingen op GEOS-gebied, het verstrekken van adviezen hen hulp bij het gebruik van GEOS en het stimuleren van gebruikers die eigen applicaties willen maken.

De volgende vraag van dhr. Janssen komt mij een beetje verdacht voor. Hij is namelijk op zoek naar een handleiding van GeoCalc. Misschien heeft dhr. Janssen dus niet de volledige *f* 89,- neergeteld voor dit GEOS-pakket. Maar laat ik niemand veroordelen zonder de feiten te kennen. Het is misschien een idee om eens bij de bovengenoemde club te informeren.

Start-diskette

Dhr. Woltering uit Didam heeft onlangs de budgetverpakking GEOS 2.0 met Nederlandse handleiding aangeschaft. Nadat hij het pakket volgens voorschrift had geïnstalleerd en de nodige backups en werkdiskettes had gemaakt, was hij van plan de volgende dag met GeoWrite te gaan beginnen. Conform de aanwijzingen had hij steeds alle diskettes met het 'close'-icon gesloten. De volgende dag weigerde de start-diskette echter te starten. Na het loaden van "geos boot" gaat er tijdens het controleren van de authenticiteit iets mis, en reset de computer. Een beeld dat alle aspirant GEOS- krakers bekend voor zal komen.

Dhr. Woltering heeft zijn defecte start-disk gecontroleerd met het standaard Commodore-programma 'disk-check'. De fout bleek een "disk ID mismatch 13" te zijn. Er is een aantal mogelijke oorzaken voor het probleem, maar de waarschijnlijkste lijkt mij, dat dhr. Woltering met verschillende gelijknamige diskettes heeft gewerkt en de ene disk uit de drive heeft gehaald, vervolgens de andere erin heeft gedaan, en toen heeft ge'close'd. De gegevens op track 18 sector 0 zijn dan overschreven met de waarden van de eerste disk. Een andere (minder waarschijnlijke) oorzaak is dat de diskette gewoon van slechte kwaliteit is, en een track gewoon de mist is ingegaan.

Een mogelijke oplossing is om met een kopieerprogramma (als dhr. Woltering daarover beschikt) een kopie te maken van een backup-diskette op de start-diskette. Als het goed is, moet alles dan weer werken. Mocht dat niet lukken dan is de enige mogelijkheid om met het pakket terug te gaan naar de verkoper.

Dit was dan weer het einde van de de GEOS-INFO rubriek, we hebben hem deze maand doorstaan zonder een enkele printervraag. Een unicum volgens mij. Vanwege een vakantiepauze is er deze maand geen GEOS-machinetaal, maar uit die hoek is nog wel het een en ander te verwachten.

Peter Boncz

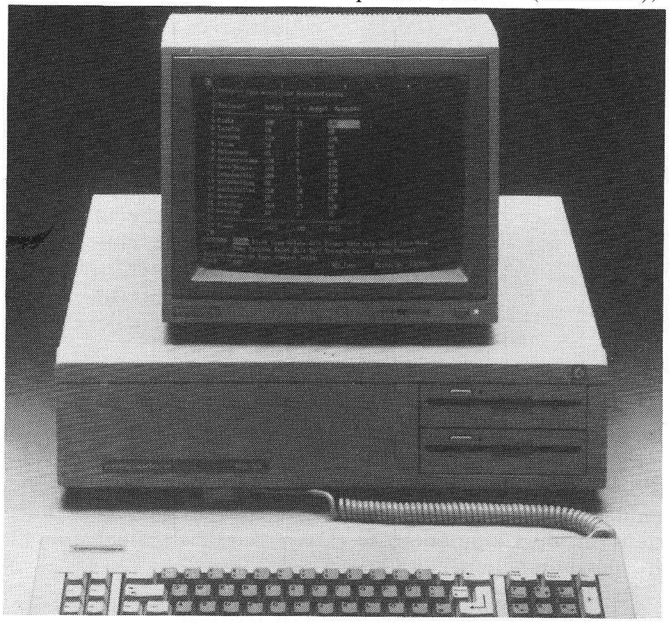


VAN C-64 NAAR PC

Het kind met het badwater weggooien?

De Commodore C-64 en C-128 behoren nog altijd tot de populairste homecomputers. Het begrip "homecomputer" is in de loop der jaren echter sterk aan veranderingen onderhevig. Eerst verstond men daar onder een hobby-apparaat, later weer een spelletjes machien, toen weer een thuiskantoor en daarna weer een speel-PC. De C-64 en later ook de C-128 hebben al deze functies prima vervuld. Hoe je het echter ook wendt of keert de serieuze toepassingen van de C- 64/128-lijn zijn tanende. Zakelijke software die enkele jaren geleden nog volop verkrijgbaar was kwijnt snel weg. Spelletjes zijn er wel nog vollop en er komen gestadig nieuwe games bij. Tegenover de C-64 en C-128 staan nu de Personal Computers (PC's). Machines die als een soort "thuiskantoor met speelse component" worden aangeprezen.

Bij talrijke PC-privé-projecten en via grootwinkelbedrijven zijn vele tienduizenden PC's in de huiskamer beland of inmiddels weer naar de zolder verhuisd. Men zegt dat als je een beetje wil meetellen het brieven typen, belasting aangeven, studiescripties maken en de boekhouding doen vanachter de PC dient plaats te vinden. Een Commodore C-64 of C-128 is daarmee in vergelijking een "kindercomputer". Waar of niet waar? We zullen de voor- en nadelen in dit artikel eens tegen elkaar afwegen. Wellicht besluit u na het lezen om er toch maar een Commodore PC bij te nemen.



Wat is een PC?

Onder de term **Personal Computer** verstaat men in het algemeen een computer voor zakelijk gebruik die men op kantoor of thuis op het eigen bureau kan zetten. Een betere term in deze zou **Desktop- computer** zijn. Zodra de term PC valt ontstaat er een sfeertje van "serieus-gebruik". Een PC is een zakencomputer en de mensen die daar achter werken zijn "echte automatiseerders". Zelfs als je daar thuis mee de brieven naar oma schrijft.

Uiteraard zal de oplettende lezer het voorgaande met een fiks korreltje zout nemen. Op de meeste kantoren zal een PC inderdaad voornamelijk zakelijk gebruikt worden. Bekende toepassingen zijn daarbij tekstverwerking, spreadsheets (calculatie), boekhouding, databases, desktoppublishing en grafische of industriële vormgeving. Daar is de PC tot een professioneel kantoorinstrument geworden.

Thuis liggen de zaken meestal anders. De PC werd via een voordelig PC-privé-project of warenhuis-aanbieding aangeschaft. Belangrijke overwegingen waren daarbij:

- Het meegaan met het automatiseringstijdperk
- De kinderen moeten er bij hun studie mee leren omgaan
- Thuisstudie: het leren omgaan met kantoor-software of een cursus volgen
- Hobby
- Statussymbool

In de praktijk kwam het daar nogal eens niet van. Een PC blijkt al snel geen stuk speelgoed, maar een ingewikkeld en soms zelfs recalcitrant stuk elektronica. Alleen de doorzetter kon er mee uit de voeten. Bovendien bleek maar al te vaak dat vele PC-taken op de huiselijke schaal net zo goed met de hand als op de computer kunnen worden uitgevoerd. Alleen de leer/studie- en hobby-aspecten blijven dan overeind. Overigens een prima reden om een PC aan te schaffen!

Tot zover het gebruik en nu de elektronica of hardware. Het gros van de PC's zijn op de Intel 80XX- of 80X86-microprocessorlijn gebaseerde computers. Deze zogenaamde IBM-compatibele PC's draaien onder het besturingssysteem MS-DOS (of het gelijkwaardige PC-DOS). Daarnaast staan de alternatievelingen zoals de op de Motorola 680X0 CPU gebaseerde Commodore Amiga's, Atari- en Applecomputers. Al deze computers beschikken over veel meer geheugen, drives (zowel diskdrives als harde schijven) en uitbreidingsmogelijkheden dan de C-64 en C-128. Er kan dus in principe zwaardere software op draaien en er kan meer mee dan bij het gewone Commodore-tje. Verder zijn er geen noemenswaardige

verschillen tussen de Home en Personal Computer.

Zakelijk gebruik

De voornaamste redenen om een PC aan te schaffen zijn de hiervoor al genoemde serieuze toepassingen. Tal van deze software- toepassingen zijn ook voor de C-64 en C-128 verkrijgbaar. De leverbare pakketten voor tekstverwerking, beperkte desktoppublishing, spreadsheets, boekhouding, tekenprogramma's en elektronische kaartenbakken voldoen veelal ruimschoots voor het huiselijk gebruik. Randapparatuur zoals printers, muizen en modems zijn ruim leverbaar. Toch loopt de veeleisende gebruiker al snel tegen de beperkingen aan.
Een van de belangrijkste beperkingen vormt de uitwisselbaarheid (**compatibiliteit**) met het kantoorgebeuren. De werknemer die de toepassingen op de zaak op zijn of haar C-64/128 wil uitwerken komt bedrogen uit. De software past domweg niet en een file thuis bewerken is er dus niet bij. Dit in tegenstelling tot de PC. De meeste MS-DOS-software draait ook op de thuis-PC. Alleen bij echt zware toepassingen kan hardware-incompatibiliteit (de computer op de zaak is veel krachtiger) optreden.
Behalve de software- is er ook hardware-incompatibiliteit tussen de C-64 of C-128 en de IBM-compatibele PC. Randapparatuur zoals printers en diskdrives blijkt veelal niet uitwisselbaar met alle gevolgen vandien.
Uit studie- en leeroogpunt is het minder verstandig om met zakelijke toepassingen op de C-64 of C-128 te werken. Hoewel de principes vaak hetzelfde zijn verschilt het gebruik dikwijls huizenhoog. Dan ben je met twee verschillend werkende pakketten bezig en dat geeft weer aanleiding tot vergissingen en extra leertijd. Wie zich thuis op een toekomstige studie, betrekking of op kantoorwerk wil voorbereiden is gewoon beter af met een PC en de voor het beoogde doel gangbare MS-DOS-software. Voor grotere zakelijke doeleinden zijn de **mogelijkheden** van de Commodore C-lijn **gewoon te beperkt**. Grote bestanden passen niet in het geheugen, de bewerkingsmogelijkheden zijn onvoldoende en alles werkt veel te traag. Waar het thuis nog best kan blijkt in het harde zakenleven, waar tijd geld is en gebruiksgemak een must betekent, C-64- of C-128-software een crime. Goed gereedschap is het halve werk en de ogen sluiten voor de beperkingen heeft geen zin.



Spelletjes

De Commodore C-64 en C-128 verslaan de PC glansrijk op twee velden: de aanschafprijs en de spelletjes. Het eerste spreekt voor zich. Meer en zwaardere hardware kost gewoon extra geld. Dat spelletjes de PC niet zo goed afgaan is minder bekend.
Een Personal Computer werd ontworpen voor puur zakelijk gebruik. Toen door de sterk dalende prijzen deze machine ook binnen hobby- en huiskamerbereik kwam ontstond de "OH-JEE-erlebnis" van "er moet ook nog vermaak bij". Dat werd verbouwen. Een kleurenkaartje en een game-adapter voor de joystick er bij zetten maakt van de PC zeker geen flitsende spelcomputer. In de praktijk vielen de weergave en de snelheid op de IBM-compatibele PC flink tegen. Het geluid was een echte aanfluiting. De Amiga vormt daarop gelukkig een zeer gunstige uitzondering.
Inmiddels zijn de technische prestaties van de IBM-kloon sterk verbeterd. EGA- en VGA-videokaarten geven met digitale game- adapters prima spelresultaten. Helaas tegen een hoge aanschafprijs en bij een relatief (in verhouding tot de C- en Amiga-lijn) klein aantal spelletjes. De meeste PC-spelletjes zijn nog van het nauwelijks te pruimen CGA-type. Gezien de stevige prijzen voor de PC in kleur met game-software wint de C-64/128 het op het gebied van spelletjes glansrijk van de Personal Computer!

De keuze

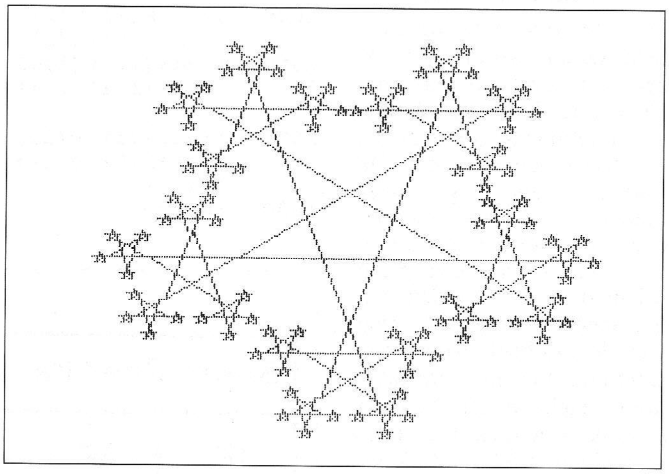
Wat nu? Met het oog op serieuze software-ontwikkelingen bij het bedrijf, een toekomstige baan, een komende studie, hobby, interesse of voor de ontwikkeling van de kinderen de C-64/128 verkopen en een Commodore PC aanschaffen? Dat valt te bezien. Qua kwaliteit en prestaties koopt u met een van de **Commodore PC's** een uitstekend produkt tegen een alleszins redelijke prijs. Wie om de bovengenoemde reden een PC aanschaft is daar zeker niet aan bekocht. Voor alleen maar spelletjes of een beetje thuisadministratie eigenlijk wel.
Het verkopen van de C-64/128 is een beetje stom. Die kunt u gewoon voor de games houden en er nog jaren plezier van hebben. Wegdoen lijkt op het spelletjeskind met het badwater weggooien!
Een alternatief voor wie zowel zwaar PC-werk als schitterende spelletjes wil spelen vormt de Amiga 2000 of 2500. Het kost wat meer maar dan krijgt u ook een machine in huis die zowel als een MS-DOS PC (van XT t/m 386SX) als een schitterende spelletjesmachine en video- en muziekcomputer kan functioneren.
Conclusie: elk PC-type heeft zo zijn eigen optimale toepassingsgebied. Voor echt zakelijk gebruik valt de Commodore PC in zijn prijsklasse niet te verslaan. Een Amiga 2000/2500 met dezelfde MS-DOS-mogelijkheden is duurder. Daartegenover staan dan weer de extra video-, muziek- en gameprestaties. De C-64 en C-128 zijn prima geschikt voor leuke gezinsspellen en een klein thuiskantoortje.

U.S.

Graphics op de 64

Deel 11: Fractals

Fractals staan de laatste jaren sterk in de belangstelling. Er zijn reeds veel boeken over geschreven en ook de publicaties in verschillende tijdschriften over dit geliefde onderwerp zijn talrijk. Niet verwonderlijk als je bedenkt dat er met zeer eenvoudige wiskundige formules, prachtige figuren op het scherm kunnen worden getoverd. Vooralsnog leek het of alleen de wat zwaardere computers als de Amiga en PC's mooie fractals op het scherm konden toveren, temeer omdat het produceren van een fractal veel rekenwerk vergt. Michel de Boer en Hylke Sprangers laten in de komende twee afleveringen van "Graphics op de 64" zien, dat er wel degelijk fractals op de 64 gemaakt kunnen worden. In dit eerste deel over fractals zal een inleiding worden gegeven op dit boeiende onderwerp. Veel wiskunde zal er overigens niet aan te pas komen.



Figuur 1

Wat zijn fractals?

Een fractal is een gecompliceerd meetkundig figuur, dat geheel door (vaak eenvoudige) wiskundige formules wordt beschreven. De meetkundige figuren worden daarbij gekenmerkt door een bepaalde zelfgelijkvormigheid. Zelfgelijkvormigheid betekent dat er bepaalde motieven in de figuur steeds terugkomen bij uitvergroting van de figuur. Als er dus een stukje van een fractal wordt uitvergroot (zoals een foto), blijkt dat stukje dezelfde motieven te bevatten als het hele plaatje. Op deze manier kan een fractal oneindig vaak uitvergroot worden. Het resultaat zal echter zijn, dat steeds de oude basisfiguur terugkomt. Als voorbeeld noemen we een boom. De stam heeft zijtakken. Als we de zijtakken bekijken, zien we dat deze zelf ook weer zijtakken heeft, enz. Het fascinerende aan fractals is, dat er met zeer simpele wiskundige formules, zeer complexe meetkundige figuren ontstaan.
In Commodore Info No.8 (dec. '89) hebben we een programma gegeven waarmee de boom van Pythagoras getekend kon worden. De boom van Pythagoras is een goed voorbeeld van een fractal. Er wordt begonnen met een vierkant met daarop een driehoek. Op de twee bovenste zijden van de driehoek wordt ook weer een vierkant met daarop een driehoek getekend. Zo kunnen we oneindig lang doorgaan met vierkanten en driehoeken tekenen. De zelfgelijkvormigheid bestaat daarin, dat elk deel van de boom weer opgevat kan worden als een boom (in een wat kleinere vorm).
Er bestaan een heleboel verschillende fractals. In deze aflevering zullen we ons richten op de wat eenvoudigere fractals. In de volgende aflevering gaan we het hebben over fractals met een wat ingewikkelder structuur, zoals de Mandelbrot- en de Julia-set.
Alle programma's zijn geschreven in Simon's Basic.

Als er ergens in de listing {pi} staat, moet daarvoor gewoon de griekse letter, links naast de restore-toets, ingetikt worden. Verder zullen we ons in deze aflevering vooral richten op voorbeelden van fractals met programma's die deze fractals op het scherm tekenen. De wiskundige theorie achter de fractals laten we een beetje voor wat ze is, en zullen we dan ook nauwelijks uitleggen.

Zeef van Sierpinski.

De eerste fractal die we gaan behandelen wordt wel de zeef van Sierpinski genoemd. Deze fractal bestaat uit een verzameling van driehoekjes, die steeds kleiner worden. De fractal is als volgt opgebouwd uit al deze driehoeken. Er wordt begonnen met het tekenen van een grote driehoek in het midden van het scherm. Aan elke zijde van de driehoek wordt een aangrenzende, kleinere driehoek getekend. Zo grenzen aan elke driehoek drie andere, kleinere driehoeken. Voor deze drie kleinere driehoeken geldt hetzelfde verhaal: Aan elke zijde grenst weer een volgende, nog kleinere driehoek. Afijn, als de computer de fractal op het scherm heeft getekend, wordt het wel duidelijker. U kunt dan goed in de figuur zien dat hetzelfde motiefje (een driehoek met daaromheen drie andere driehoeken) steeds weer terugkomt op een andere schaal. Dit verschijnsel hadden we zelfgelijkvormigheid genoemd. De fractal is genoemd naar de Poolse wiskundige Waclaw Sierpinski (1882 - 1969). Het is wel duidelijk waarom de figuur wordt geassocieerd met een zeef. We geven nu een programma waarmee de zeef van Sierpinski getekend kan worden.

```
5    rem zeef van
        Sierpinski
10   hires 0,1
20   l(1)=100:x(1)=160
     :y(1)=100-l(1):i=
30 gosub 100
40 goto 40
100 if i>5 then
     i=i-1:return
110 gosub 1000
120 l(i+1)=l(i)/2:x(i+1
     )=x(i)-l(i+1):y(
     i+1)=y(i):i=i+1
     :gosub 100
130 l(i+1)=l(i)/2:x(i
     +1)=x(i)+l(i+1):y(
     i+1)=y(i):i=i+1
     :gosub 100
140 l(i+1)=l(i)/2:x
     (i+1)=x(i)+l(i+1):
     y(i+1)=y(i)+
     l(i):i=i+1:gosub
     100
150 i=i-1:return
1000    line x(i),y(i),
        x(i),y(i)+l(i),1
1010    linex(i),y(i)+l(i)
        ,x(i)+l(i),y(i)+
        l(i),1
1020    line x(i),y(i),
        x(i)+l(i),y(i)+
        l(i),1
1030    return
```

Het programma ziet er op het eerste oog nogal moeilijk uit. Dit komt omdat we de programmeertechniek recursie hebben gebruikt. Recursie is een programmeertecniek waarbij een routine zichzelf aanroept. In programmeertalen zoals Pascal kan met vrucht van deze tecniek gebruik gemaakt worden. In Basic wordt het echter al gauw een onoverzichtelijk zootje. Omdat het niet onze bedoeling is om hier een cursus gevorderd programmeren te geven, zullen we slechts kort stil staan bij de werking van dit programma.
Op de regels 1000 en verder staat een subroutine die een driehoek op het scherm tekent. De recursieve routine begint op regel 100. Deze routine tekent eerst een driehoek door in regel 110 de subroutine op regel 1000 aan te roepen. Daarna roept deze routine zichzelf drie keer aan in de regels 120 tot en met 140. Elke aanroep is voor het tekenen van een van de drie kleinere driehoeken rondom de net getekende driehoek. Voor elk van deze drie kleinere driehoeken wordt de routine ook weer drie keer aangeroepen, namelijk voor het tekenen van de drie nog kleinere driehoeken rondom deze driehoek. Zo kunnen we in principe steeds dieper gaan. De variabele i geeft aan hoe diep we zitten met het tekenen van steeds kleinere driehoeken. In regel 100 wordt steeds getest hoe diep we zitten. Als we op een diepte van 6 zijn aangeland, gaat de computer niet meer verder met het tekenen van nog kleinere driehoeken, maar springt terug.

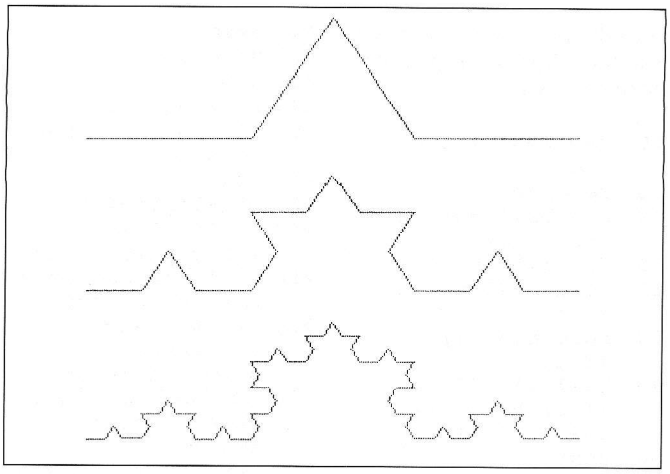
Ster Fractals

ap	ar	le	dp
5	2	40	4
4	1	60	5
7	3	90	3
4	0.6	35	5
3	1	70	5
8	5	78	4
6	0.25	20	4
7	30	3	3
6	39	4	4

Figuur 2

Ster fractals

Hoewel fractals feitelijk zuiver wiskundige objecten zijn, vinden ze hun oorsprong vaak in de natuur. Het eenvoudigste voorbeeld hiervan is de spiraal. De spiraal kan wel opgevat worden als de oerfractal. In de natuur vinden we de spiraal terug bij bijvoorbeeld de vorm van schelpen. Ook de structuur van de longen doet sterk aan fractals denken. Als laatste voorbeeld geven we de ijsbloem, die 's winters vaak op de ramen is te vinden. Deze heeft vaak een ster-vormige structuur. In deze paragraaf zullen we het hebben over ster-fractals.
In figuur 1 is een voorbeeld te vinden van een ster fractal. U kunt in deze figuur goed zien hoe ze is opgebouwd. Het basis figuurtje is, hoe kan het ook anders, een ster. De eerste (davids-) ster wordt groot in het midden van het beeld getekend. Op elk van de 5 hoekpunten van de ster, staat weer een nieuwe ster. Deze vijf kleinere sterren hebben aan elk van de vier vrije hoekpunten weer een nieuwe, nog kleinere ster. Dit gaat weer oneindig door. Ook in dit figuur is de zelfgelijkvormigheid weer duidelijk te zien.
We zullen nu bekijken hoe we een ster op het beeld kunnen tekenen. Merk op dat de hoek tussen twee lijnen die elkaar snijden in een hoekpunt van de ster, steeds gelijk is. Omdat we de hoeken kennen, kunnen we, uitgaande van een hoekpunt, de overige hoeken eenvoudig met een sinus en een cosinus berekenen. We geven nu een programma waarmee ster fractals getekend kunnen worden.



Figuur 3

```
5   rem ster fractals
6   input"aantal punten";ap
7   input"aantal maal rond";ar
8   input"beginlengte lijn";le(1)
9   input"diepte";dp
10  hires 0,1:i=1:dh(1)=360*ar/ap
20  x1(1)=160:y1(1)=140:h(1)=dh(i)/2+90
30  gosub 100
40  goto 40
100 t(i)=0:if i>dp then i=i-1:return
120 t=h(i)*{pi}/180
130 x2(i)=int(le(i)*sin(t)+x1(i)+.5)
135  y2(i)=int(le(i)*cos(t)+y1(i)+.5)
140 line x1(i),y1(i),x2(i),y2(i),1
145 t(i)=t(i)+1
146 if i>1 and t(i)=ap then i=i-1:return
147 x1(i)=x2(i):y1(i)=y2(i):i=i+1
150 x1(i)=x2(i-1):y1(i)=y2(i-1):le(i)=le(i-1)/2.5:h(i)=h(i-1):dh(i)=-dh(i-1)
155 gosub 100:h(i)=h(i)+dh(i)
157 if t(i)=ap then i=i-1:return
170 goto 120
```

Ook dit programma is weer geschreven met behulp van recursie. We zullen het programma niet gaan uitleggen. Wel zullen we duidelijk maken wat voor een figuren er allemaal met dit programma gemaakt kunnen worden.

De afbeelding in figuur 1 is namelijk niet de enige ster fractal die gemaakt kan worden. Ster fractals kenmerken zich door een basisfiguur, zoals bijvoorbeeld een vierkant, een zesdhoek, of een davidsster. Dit basisfiguur wordt midden op het beeld getekend en op elk van de hoekpunten van dit basisfiguur wordt vervolgens weer zo'n figuurtje getekend, alleen dan veel kleiner. Als voorbeeld nemen we nu een zeshoek. Eerst wordt er een grote zeshoek op het beeld getekend. Vervolgens wordt er op elk van de zes hoekpunten weer een zeshoek gezet. Dit gaat zo door.

Met het gegeven programma kunnen dergelijke fractals gemaakt worden. We zullen nu uitleggen hoe het basisfiguurtje met het programma in de computer ingegeven kan worden.

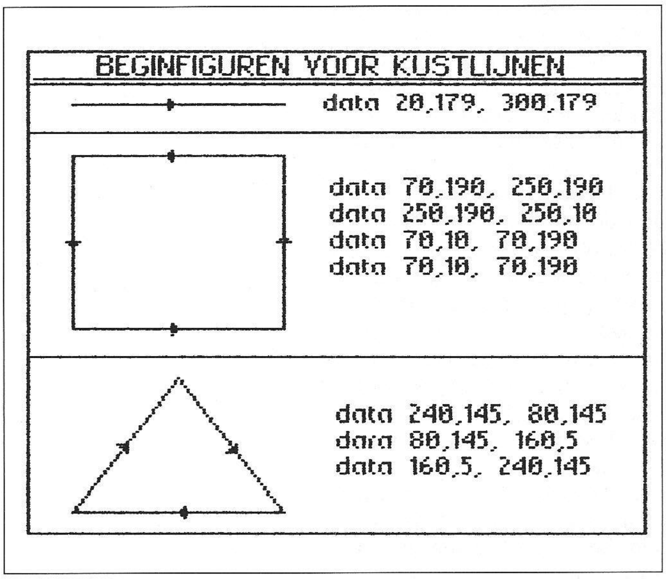
U kunt zien dat er vier variabelen in het begin van het programma ingegeven moeten worden. De laatste variabele, de diepte, spreekt voor zich: deze geeft aan hoe diep er gegaan wordt met steeds maar kleinere figuurtjes tekenen. De variabele "begin lengte lijn" spreekt ook voor zich: deze geeft de lengte aan van de afzonderlijke lijnen van het eerste basisfiguurtje. Als het basisfiguurtje bijvoorbeeld een vierkant is, geeft deze variabele de lengte van de zijden aan van het eerste vierkant. Met de eerste twee variabelen kunt u het beginfiguurtje vastleggen. De eerste variabele geeft daarbij aan hoeveel hoekpunten het figuurtje heeft. Bij een zeshoek moet u hier dus het getal 6 ingeven. De tweede variabele "Aantal maal rond", geeft aan hoe deze hoekpunten zijn verdeeld over het figuurtje. Als we hier 1 ingeven, zullen de hoekpunten keurig verdeeld worden over 360 graden. Dit betekent dat er bij vier hoekpunten een vierkant onstaat, en bij 6 hoekpunten een zeshoek. Wordt er nu bijvoorbeeld 2 ingegeven, dan worden de hoekpunten verdeeld over 360*2 = 720 graden. Als er bijvoorbeeld 5 hoekpunten zijn opgegeven, betekent dat, dat de twee lijnen die elkaar kruisen in een hoekpunt, onder een hoek van 720/5 = 144 staan. In dit geval wordt dan de afbeelding in figuur 1 verkregen. U kunt dus met de tweede variabele bepalen wat de hoek is tussen de twee lijnen die elkaar kruisen in een hoekpunt.

U kunt het beste maar wat gaan experimenteren met het programma: de werking wordt dan vanzelf duidelijk. In figuur 2 staat een tabel met leuke waarden voor het programma.

Kustlijnen

Stelt u zich het volgende scenario voor. U bent een ruimtereiziger afkomstig van een planeet in een verre uithoek van de paardekopnevel. Na vele lichtjaren te hebben gereisd komt u in de buurt van de planeet Aarde. Van grote afstand neemt u de kustlijn van Engeland waar en u probeert te meten hoe lang deze is. Naarmate u de Aarde dichter nadert worden er steeds meer details van de kustlijn zichtbaar. Stukken die eerst recht leken, blijken allerlei uitstulpingen te bevatten, waardoor een nieuwe

meting een grotere lengte oplevert. Dit gaat zo door totdat u geland bent. De kustlijn wordt echter nog langer, als we deze met een microscoop gaan bekijken, er worden dan nog meer uitstulpingen zichtbaar. We kunnen zo doorgaan tot in het oneindige. Omdat de lengte van de kust toeneemt naarmate de waarneming gedetailleerder wordt, kunnen we concluderen dat de lengte oneindig is!



Figuur 4

Dit gedachtenexperiment werd in de zestiger jaren door Benoit Mandelbrot, een wiskundige van Poolse komaf, uitgevoerd. Mandelbrot merkte ook op dat de vorm van de kustlijn bij elke waarneming vrijwel gelijk was. Dus de kustlijn bezit een zekere mate van gelijkvormigheid. Hiermee was de fractal geboren.
We hebben een programma geschreven, dat gebaseerd is op bovenstaand experiment. Met dit programma kan een scala van wiskundige kustlijnen gegenereerd worden. In figuur 3 is een bekende kustlijn afgebeeld, de Kochkromme (genoemd naar de wiskundige Koch). De bovenste figuur toont de minst gedetailleerde waarneming van de kromme. Daaronder is de kromme een keer vergroot. Nog meer vergroten geeft de onderste kromme. De opbouw van de kromme is simpel. We beginnen met een lijnstuk. Dit lijnstuk vervangen we door de bovenste kromme; deze kromme zullen we in het vervolg de basiskromme noemen. Hierin vervangen we vervolgens elk lijnstuk door een verkleinde versie van de basiskromme.

Zo onststaat de tweede kromme. Vervolgens vervangen we de lijnstukken hierin weer door nog kleinere versies van de basiskromme. In theorie kunnen we zo tot het einde der dagen doorgaan. In de praktijk zijn de lijnstukken na een keer of vijf zo klein, dat we het proces niet nog een keer kunnen uitvoeren.
Het is ook leuk om niet met een lijnstuk te beginnen maar met een driehoek. We maken dan met het zojuist beschreven proces van elke zijde van de driehoek een Kochkromme. Zo ontstaat het Kocheiland. Andere variaties op de Kochkromme ontstaan door een andere vorm voor de basiskromme te nemen. Bijvoorbeeld een lijn met een vierkant erop in plaats van een driehoek. Met het onderstaande programma kunnen alle mogelijke kustlijnen gegenereerd worden.

```
10  rem kustlijnen
20  rem
30  read ab
40  dim bl(3,ab),rh
    (ab),l(ab)
50  for i=1 to ab
60  read bl(0,i),bl
    (1,i)
70  read bl(2,i),bl
    (3,i)
75  bl(1,i)=199-bl
    (1,i):bl(3,i)=
    199-bl(3,i)
80  next
90  read am
100 dim mp(1,am),h
    (am),rl(am)
110 for i=1 to am-1
120 read mp(0,i),mp
    (1,i)
130 next
140 mp(0,am)=1
150 input"vergrot
    ing";d
160 dim w(d)
170 for i=1 to ab
180 lx=bl(2,i)-bl
    (0,i):ly=bl(3,i)
    -bl(1,i)
190 if lx=0 then
    h={pi}/2:else:
    h=atn(abs(ly/lx))
200 gosub 480:rh(i)=h
210 l(i)=sqr(lx^2+ly
    ^2)
220 next
230 for i=1 to am
240 lx=mp(0,i)-mp
    (0,i-1):ly=mp(1,i)
    -mp(1,i-1)
250 if lx=0 then
    h={pi}/2:else:
    h=atn(abs(ly/lx))
260 gosub 480:h(i-1)
    =h
270 rl(i-1)=sqr
    (lx^2+ ly^2)
280 next
290 hires 0,15
310 for i=0 to am^d-1
320 for j=1to ab
330 h=rh(j):rl=1
340 for k=0 to
    d-1:h=h+h(w(k)):
    rl=rl*rl(w(k)):
    next
350 bl(2,j)=bl(0,j)+
    l(j)*rl*cos(h)
360 bl(3,j)=bl(1,j)+
    l(j)*rl*sin(h)
370 line bl(0,j),199
    -bl(1,j),bl(2,j)
    ,199-bl(3,j),1
380 bl(0,j)=bl
    (2,j):bl(1,j)=bl
    (3,j)
390 next j
400 gosub 430
410 next i
420 goto 420
430 q=0
440 w(q)=w(q)+1
450 if w(q)<>am then
    return
460 w(q)=0
470 q=q+1:goto440
480 if lx>=0 and
    ly>=0 then return
490 if lx>=0 and ly<0
    then h=-h:return
500 if lx<0 and
    ly>=0then
    h={pi}-h:return
510 if lx<0 and ly<0
    then h={pi}+h:
    return
520 rem aantal lijnen
    beginfiguur
530 data 1
540 rem begin- en
    eindpunten van
    lijnen
550 data 20,179,
    300,179
560 rem aantal lijnen
    van basiskromme
570 data 4
580 rem coordinaten
    van eindpunten
590 data .333,0
591 data .5,.287
592 data .667,0
```

Als u het programma start, vraagt het om de vergrotingsfactor. Factor 1 geeft de basiskromme. Factor 2 geeft de kromme na een keer vergroten, enz. Na het invoeren van de factor, wordt de Kochkromme getekend.
Door de data vanaf regel 520 te veranderen, kunnen nieuwe kustlijnen ontworpen worden. Het getal in regel 520 geeft aan uit hoeveel lijnen het beginfiguur bestaat. Bij de Kochkromme beginnen we met 1 lijnstuk. Het Kocheiland begint met een driehoek; het beginfiguur heeft dan 3 lijnen. Vanaf regel 540 staan de coordinaten van het begin- en eindpunt van elke lijn. Als er meerdere lijnen zijn, dan moet er voor elke lijn een datastatement worden opgenomen. In figuur 4 staan een aantal voorbeelden van beginfiguren, met daarbij de datastatements die in het programma opgenomen moeten worden. De eerste twee getallen in een datastatement zijn de coordinaten voor het beginpunt van

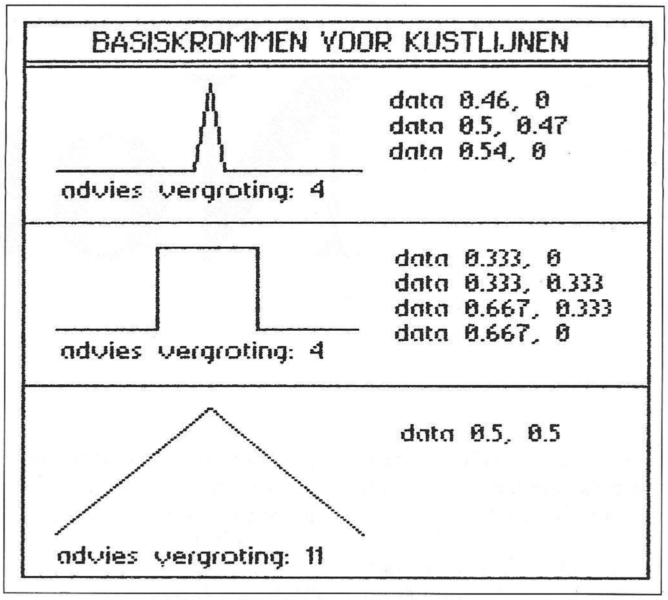
een lijn en de laatste twee corresponderen met het eindpunt van deze lijn. De pijlen in de figuren geven aan in welke richting de kustlijn getekend wordt. Deze richting is van belang. Als de Kochkromme van links naar rechts wordt getekend, dan ontstaat de kromme, zoals in figuur 3. Tekenen we echter van rechts naar links, dan staat de kromme op zijn kop. Als we met een lijn als beginfiguur starten, geeft het veranderen van de tekenrichting geen wezenlijk andere kustlijn. Bij beginfiguren met meerdere lijnen kunnen er echter verrassende beelden ontstaan. U kunt de tekenrichting van een lijn veranderen door het de coordinaten van het begin- en eindpunt in het datastatement, behorende bij de lijn, te verwisselen.

Nu moet nog de vorm van de basiskromme worden ingevoerd. Regel 570 geeft aan uit hoeveel lijnen de basiskromme bestaat. Bij de Kochkromme zijn dit 4 lijnen. De regels vanaf regel 590 bepalen de vorm van de basiskromme. In het programma is het beginpunt van de lijn, waarop de kustlijn wordt getekend, als punt (0,0) en het einpunt als (0,1) gedefinieerd. Het beginpunt van het eerste lijnstuk van de basiskromme is nu (0,0). In regel 590 zetten we nu de coordinaten van het eindpunt van dit lijnstuk. Voor de Kochkromme is dit (0.333, 0). Dit zijn tevens de coordinaten van het begin van het tweede lijnstuk. Vervolgens zetten we de coordinaten van het eindpunt van het tweede lijnstuk in een datastatement. Dit doen we voor alle lijnstukken, behalve voor de laatste; het eindpunt van het laatste lijnstuk is namelijk (1,0) en is al bekend bij het programma.

In figuur 5 staan een aantal voorbeelden van basiskrommen, die u kunt proberen om leuke kustlijnen te genereren. Bij elke basiskromme staan de datastatements vermeld die in het programma moeten worden opgenomen.

Het klinkt allemaal misschien wat ingewikkeld, maar na enig experimenteren met de voorbeeldwaarden zult u het creeeren van prachtige kustlijnen wel onder de knie hebben. Veel plezier ermee.

Hylke Sprangers & Michel de Boer



Figuur 5

Nordic

Er zijn vele cartridges ontwikkeld die de Commodore 64 uitbreiden met extra Basic instructies en handige utilities zoals disk turbo's en monitoren. De laatste jaren is er een geheel nieuwe generatie cartridges ontworpen. Deze cartridges maken het mogelijk om lopende programma's te bevriezen en vervolgens een backup op diskette te maken of een screen dump op de printer af te drukken. Hierna kan het programma dan weer opgestart worden. Bekende cartridges uit deze generatie zijn de Power Cartridge, de Expert Cartridge en de Final Cartridge. Een nieuwe cartridge in deze generatie is de Nordic Power.

De Nordic Power is een produkt van Data & Electronics uit Venlo. De Nordic Power breidt het Basic uit, kan programma's bevriezen en vertragen, versnelt disk operaties en bevat een toolkit met allerlei handige utilities. Wij hebben deze functies van de Nordic Power voor u getest.
De Nordic Power is een klein rood kastje dat in de cartridge expansie poort moet worden gestoken. Aan de achterkant van de cartridge bevinden zich twee drukknoppen. De ene is een resetknop, de andere dient voor het bevriezen van programma's. Het is ook mogelijk om de Nordic Power samen met de Commodore 128 te gebruiken. Deze wordt dan echter automatisch in 64 mode gezet.

Als de Nordic Power is aangesloten en de computer wordt aangezet, verschijnt er een menu op het scherm. Vanuit dit menu kan men uit de volgende opties kiezen:

° Reset
° Reset en geheugen wissen
° Power Basic
° Toolkit
° Monitor

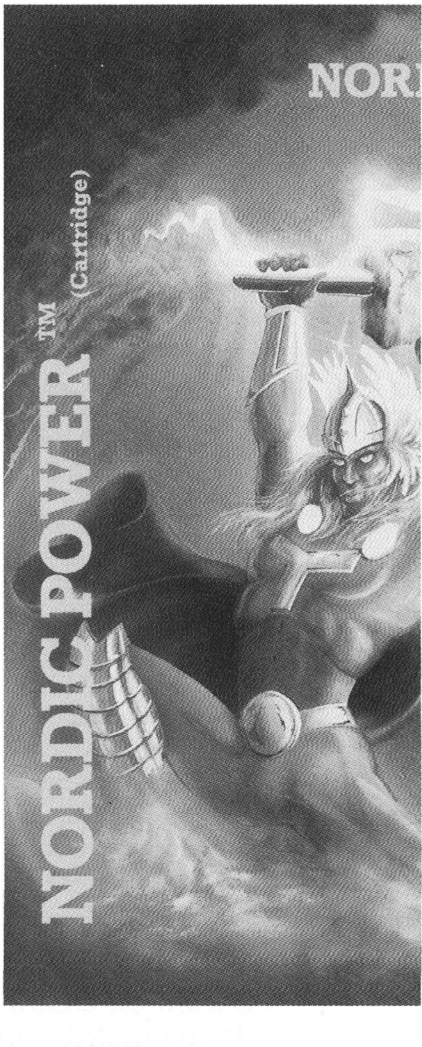
Dit zelfde menu verschijnt ook als men op de resetknop drukt.

Power Basic

Power Basic biedt een aantal nuttige instructies die het ontwikkelen van eigen programma's vergemakkelijken. We geven hieronder een greep uit deze instructies:
OLD herstelt een programma dat is verdwenen door NEW of resetten van de computer.
RENUMBER hernummert de regelnummers van een programma. Daarbij worden ook de GOTO en GOSUB instructies meeveranderd.
PLIST drukt de listing van een programma, dat op diskette staat, af op de printer zonder dat het programma daarbij in het geheugen geladen wordt.
Verder kan met een druk op een van de functie toetsen een aantal commando's worden uitgevoerd. Enkele daarvan zijn RUN, LIST, het tonen van een disk directory zonder geheugenverlies en het starten van de machinetaal monitor.

Disk en tape operaties

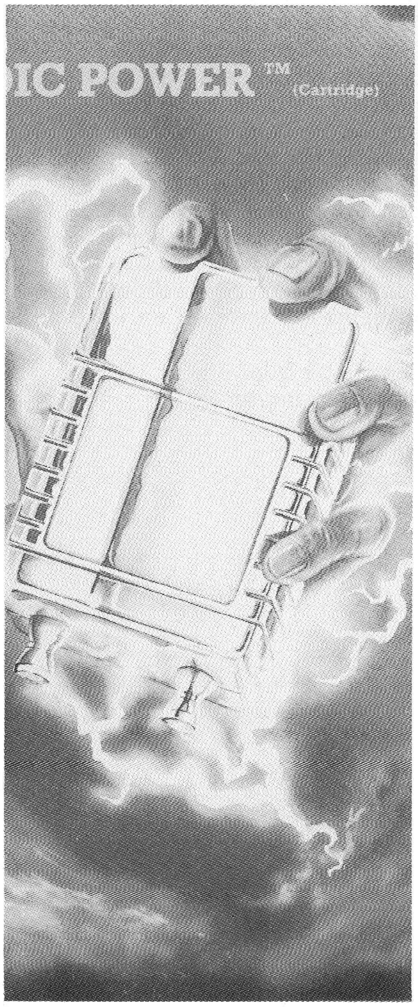
Power Basic bevat ook een disk en tape turbo. De disk turbo versnelt het laden en saven van progamma's met maar liefst een factor 15 tot 20. De programma's worden daarbij niet in een speciaal formaat weggeschreven. Alle programma's kunnen dus met de diskturbo worden geladen. De tape turbo is compatible met alle bestaande tape turbo's. De beide turbo's kunnen files met een maximum lengte van 249 blocks verwerken.
Ook het formateren van een diskette wordt aanzienlijk versneld. De tijd dat u tijdens het formateren rustig een kopje koffie kon drinken, is met de komst van Nordic Power voorbij. Het formateren duurt minder dan 20 seconden.
Een interresante optie bij het formateren is het formateren van 40 tracks. Normaal worden er 35 tracks op een diskette geformateerd. Er is dan echter nog ruimte voor 5 extra tracks die niet benut worden. Met de Nordic Power kunnen deze extra tracks ook geformateerd worden. De opslagcapaciteit van een diskette neemt daardoor toe van 664 naar 749 blocks. Het is dan echter niet meer mogelijk om programma's die zich op tracks 36-40 bevinden zonder de Nordic Power in te lezen.
Met eenvoudige twee-letter afkortingen kunnen DOS commando's worden gegeven. Het commando kanaal 15 hoeft daarbij niet meer geopend en gesloten te worden.



Power

Toolkit

De toolkit bevat een aantal copieer utilities, een cruncher en een sprite editor.
Met de copieer utilities kunnen zowel losse files als hele diskettes gecopieerd worden. Het copieren van ee hele diskette geschiedt in minder dan drie minuten tijd.



Met de cruncher kunnen sommige lange programma's ingekort worden.
De sprite editor is een editor waarmee zowel single als multi color sprites ontworpen kunnen worden.
Verder biedt de toolkit de mogelijkheid om een disk turbo op de diskette te plaatsen. Deze turbo wordt in de directory track gezet, zodat er geen opslagruimte verloren gaat. Als men vervolgens deze turbo inleest, verschijnt de directory van de diskette op het scherm. Met behulp van de cursortoetsen kan dan een file worden uitgekozen. Deze file wordt dan ongeveer zes maal sneller ingelezen dan normaal.

Monitor

De monitor van de Nordic Power werkt fantastisch. In de monitor zijn een machinetaal-, een floppy- en een diskmonitor tot een geheel geintegreerd.
De machinetaal monitor beschikt over alle gebruikelijke commando's, zoals disassembleren, zoeken en geheugendump. Een zeer handig commando is het commando om een Basic instructie uit te voeren. Hierdoor kan vanuit de monitor even een listing bekeken worden of een berekening uitgevoerd worden. Alle commando's kunnen hun uitvoer zowel naar het beeldscherm als de naar printer sturen. Met de machinetaal monitor is het volledige 64K RAM geheugen bereikbaar.
Met de disk monitor kan het geheugen van de disk drive bekeken en veranderd worden. Hiervoor zijn alle commando's van de machinetaal monitor te gebruiken. Doordat de machinetaal en de disk monitor geintegreerd zijn, kunnen er stukken geheugen van de disk drive naar het geheugen van de computer gecopieerd worden en vice versa.
De floppy monitor dient voor het bekijken en veranderen van de inhoud van een diskette. Met de floppy monitor kan een block van de diskette in het geheugen van de computer worden gelezen. Deze kan vervolgens bewerkt worden met de machinetaal en monitor. Daarna kan het block weer teruggeschreven worden op de diskette. Door deze manier van werken is de floppy monitor een stuk krachtiger dan de meeste losse floppy monitoren. Deze beschikken namelijk meestal alleen maar over een commando om veranderingen aan te brengen in een hexadecimale dump van een block.

Freezer

De freezer is karakteriserend voor de nieuwe generatie cartridges. Hiermee kunnen lopende programma's bevroren worden. Als een programma eenmaal bevroren is kunnen er allerlei functies worden uitgevoerd. Er kunnen screendumps gemaakt worden, het programma kan gecopieerd worden, sprite collisions kunnen uitgeschakeld worden en nog veel meer. Daarna kan de uitvoering van het programma weer herstart worden.
Het bevriezen van een programma gebeurt door de freeze knop achterop de cartridge in te drukken. Zeer eenvoudig dus. Helaas ging het ons niet zo gemakkelijk af. Het eerste programma dat wij probeerden te bevriezen, sloeg hopeloos vast; het freeze menu verscheen niet zoals in de handleiding vermeld stond. De handleiding vermeldde wel dat het nuttig was om de freeze knop maar kort aan te tippen, daar er anders de mogelijkheid bestond dat de freeze functie niet correct werkte. Maar helaas, ook dat mocht niet baten. Na nog een aantal pogingen onder luid gevloek is het ons toch gelukt om een paar programma's te bevriezen.

Zodra de freeze knop wordt ingedrukt, wordt de uitvoering van het programma onderbroken en verschijnt het freeze menu. Vanuit dit menu kunnen de volgende opties met behulp van de functie toetsen gekozen worden:

- F1: backup
- F3: utility menu
- F5: editor menu
- F7: monitor
- F8: programma herstarten

Backup

Deze functie maakt een copie van het hele RAM geheugen op disk op of tape. Voordat

het programma weggeschreven wordt, wordt het eerst gecomprimeerd met de cruncher, zodat het programma zo min mogelijk ruimte in beslag neemt. Als het programma later weer wordt ingelezen, dan start het precies op het punt waarop het bevroren is. Op deze manier is het mogelijk om een backup te maken van een programma dat beveiligd is tegen copieren.
De backup functie is ook handig te gebruiken bij het spelen van adventure spelen, waarbij de spelresultaten niet tussentijds gesaved kunnen worden, waardoor u telkens helemaal opnieuw moet beginnen als u dood bent. Met de backup functie kunt u op een bepaald punt een copie maken van het programma maken, zodat u altijd vanaf dit punt verder kunt spelen.
Backups die op deze manier gemaakt worden, kunnen later zonder gebruik van de cartridge ingelezen worden.

Utility menu

Het utility menu biedt een aantal leuke opties voor video game freaks. Zo kunnen de sprite collisions tussen sprites onderling en tussen sprites en achtergrond graphics kunnen worden uitgeschakeld. Hierdoor kunt u uw starfighter moeiteloos door de vijandige ruimte manouvreren; u vliegt letterlijk dwars door de vijandige linies heen. Deze optie werkt alleen in spelletjes die gebruik maken van de sprite collision registers. Programma's die sprite botsingen detecteren door middel van het vergelijken van de verschillende sprite posities, zijn immuun voor de optie.
Ook kunt u uw overbelaste duimspier wat rust geven, door de automatisch-vurenoptie te kiezen. U houdt relaxed uw duim op de vuurknop en de Nordic Power doet de rest. Een andere joystick functie is joystick omschakel functie. Hiermee wordt een eind gemaakt aan het steeds in een andere poort pluggen van de joystick, omdat het ene programma met joystick poort 1 werkt en het andere met poort 2. Met deze functie kunnen programma's die met poort 1 werken, veranderd worden, zodat ze met poort 2 werken.
Met de pokes-functie kunnen zogenaamde trainer pokes worden ingevoerd, tijdens het lopen van het programma.
De vertragings-functie zou het mogelijk moeten maken, om programma's te vertragen. Het is ons echter niet gelukt om deze functie succesvol uit te voeren. Al onze pogingen om een programma te vertragen resulteerde in het vastslaan van het programma. Een iets te rigoreuze vertraging, naar onze mening.

Editor menu

Vanuit het editor menu kan uit een aantal opties om het beeldscherm te bewerken, gekozen worden.
Een high resolution beeld, dat op het moment van het bevriezen op het scherm stond, kan op diskette worden gezet. Hiervoor zijn een aantal verschillende formaten beschikbaar. Een aantal formaten zijn: HI-EDDI, KOALA PAINTER, BLAZING PADDLES en ADVANCED ART STUDIO. Een beeld wat bijvoorbeeld weggeschreven is in KOALA PAINTER-formaat kan later verwerkt worden met behulp van KOALA PAINTER. Dit beeld kan tenslotte weer worden teruggeplaatst in het bevroren programma.
Een andere optie is het maken van een screen dump. Hiervoor zijn er twee verschillende mogelijkheden. Een scherm dat uit louter tekst bestaat kan op de printer worden afgedrukt in tekst mode. High resolution schermen en schermen, die zijn opgebouwd uit zelf-gedefinieerde karakters, kunnen worden afgedrukt in high resolution mode. Kleuren worden daarbij weergegeven in grijstinten. De duidelijkheid van de afdruk is afhankelijk van de kleurencombinaties. Sommige kleurencombinaties geven namelijk niet zo'n goed contrast. Als u echter over een EPSON kleurenprinter beschikt, dan kan er ook een screen dump in kleur worden gemaakt.
Met de dia show optie kunt u high resolution beelden uit verschillende programma's op diskette saven en samenvoegen tot een dia show. Als deze dia show later gestart wordt, worden alle gesavede beelden achter elkaar getoond. Leuk voor een demonstratie.
Als laatste probeerden we de sprite editor. Dit is dezelfde sprite editor als in de hiervoor beschreven toolkit. Omdat we nu een programma hebben bevroren, kunnen we met de sprite editor het geheugen afzoeken naar de sprites van het programma. Deze kunnen dan veranderd worden. Met de sprite editor en de beeldscherm functies kunt u dus de graphics van uw favoriete spel naar eigen smaak aanpassen. Vervolgens kan met de backup functie het veranderde spel op diskette worden gezet.

Herstarten

Na een aantal opties te hebben uitgevoerd, moet het bevroren programma natuurlijk weer worden opgestart. Een druk op de F8-toets start het programma weer op. Helaas gaf het opstarten net als het bevriezen weer de nodige problemen. Sommige programma's lieten het na het opstarten afweten. Bij andere programma's veranderde een paar karakters van kleur. Het programma deed het dan wel zonder problemen.

Handleiding

De bijgeleverde handleiding is geheel duitstalig. Op sommige punten is de handleiding wat slordig. Zo staat er bijvoorbeeld dat er met het commando 'backslash gevolgd door een naam' een programma op disk kan worden gesaved. Op het toetsenbord van de Commodore 64 is echter geen backslash te vinden.
De handleiding had naar onze mening iets uitgebreider en lezersvriendelijker mogen zijn; dit is echter een euvel waar de meeste handleidingen aan lijden. Desondanks zijn alle functies voldoende beschreven.

Conclusie

De Nordic Power is een cartridge met vele interessante mogelijkheden. Na enig stoeien met de verschillende functies, heeft men het werken met de cartridge vrij snel onder de knie.
Het concept is goed doordacht. Power Basic, de toolkit en de diverse monitoren werken prima. Een groot minpunt is echter dat de robuustheid van de freezer nogal wat te wensen over laat. Bij sommige programma's lukte het bevriezen pas na een paar pogingen en bij veel andere programma's lukte het helemaal niet. Wij hopen dat dit probleem snel verholpen wordt, daar het werken met de freezerfuncties, zoals het maken van een backup of een screendump, vrijwel onmogelijk is.

Michel de Boer &
Hylke Spranger

Tips & Trucs 64

In de Commodore 64 schuilen vele geheimen. Zo hebben we een aantal afleveringen terug laten zien hoe de borders uitgeschakeld konden worden en staat in deze aflevering van Commodore Info, in het artikel 'Graphics op de 64', een prachtige scrollmethode beschreven, waarmee hele high resolution schermen gescrolld kunnen worden. Niet alleen de video chip, maar ook de 6510 micro processor heeft geheimen: instructies die niet in de instructie set beschreven staan.

		MNEMONICS							
		AAX	DCP	ISC	LAX	RLA	RRA	SLO	SRE
ADRESMODE	zero-page	$87	$C7	$E7	$A7	$27	$67	$07	$47
	zero-page,x		$D7	$F7		$37	$77	$17	$57
	zero-page,y	$97			$B7				
	absolute	$8F	$CF	$EF	$AF	$2F	$6F	$0F	$4F
	absolute,x		$DF	$FF		$3F	$7F	$1F	$5F
	absolute,y		$DB	$FB	$BF	$3B	$7B	$1B	$5B
	(indirect,x)	$83	$C3	$E3	$A3	$23	$63	$03	$43
	(indirect),y		$D3	$F3	$B3	$33	$73	$13	$53
VLAG	zero	?	?	?	?	?	?	?	?
	negative	?	?	?	?	?	?	?	?
	carry		?	?		?	?	?	?
	overvlow			?			?		

Secret Opcodes

Figuur 1

De 6510 micro processor is het hart van de Commodore 64. Deze chip voert de machinetaal instructies van een machinetaal programma uit. Elke machinetaal instructie heeft een bepaalde code, de zogenaamde opcode (operation code). Instructies met meerdere adresseringsmodes (bijvoorbeeld absolute en zero-page,x) hebben voor elke adresseringsmode een andere opcode. De opcode voor de instructie CLC is bijvoorbeeld $18 en de opcode voor JMP met absolute adressering is $4C. De 6510 haalt steeds een opcode uit het geheugen. Vervolgens decodeert hij deze opcode en voert de juiste acties uit, zoals het wissen van de carry flag (bij CLC) of het veranderen van de program counter (bij JMP). Deze reeks handelingen wordt continu herhaald.
De opcode van een instructie past precies in een byte. Er zijn dus 256 verschillende opcodes. Als we de instructie set van de 6510 bestuderen, blijkt echter dat er minder dan 256 opcodes gebruikt worden om alle instructies te coderen. De ongebruikte opcodes zullen we in het vervolg secret opcodes noemen. Wat gebeurt er nu als de 6510 een van deze secret opcodes tegenkomt bij het uitvoeren van een machinetaal programma? Deze opcode zal net als alle andere opcodes gedecodeerd worden en afhankelijk van het resultaat van het decoderen zal er een bepaalde actie ondernomen worden. Kortom, er moeten instructies zijn die corresponderen met deze opcodes, maar die niet in de instructie set beschreven staan. In deze aflevering zullen we deze 'geheime' instructies beschrijven. Er zijn acht verschillende instructies. Hieronder zullen we van elk van deze acht instructies kort beschrijven wat de functie is, welke vlaggen beinvloed worden en welke adresseringen mogelijk zijn.

AAX

De instructie AAX (And Accumulator and X-register) voert een logische AND uit op de waarden van de accumulator en het X-register. De uitkomst van deze bewerking wordt opgeslagen in het gespecificeerde geheugenadres. De waarden van de accumulator en het X-register worden niet aangetast door AAX. AAX beinvloedt op vreemde wijze de Zero- en Negative-flag. Deze vlaggen worden namelijk afhankelijk van de waarde van het Y-register gezet. Als deze waarde 0 is, wordt de Zero-flag gezet en als bit 7 van de waarde aan staat, wordt de Negative-flag gezet.
Adressering: Zero-page; Zero-page,Y; Absolute; (Indirect,X)

DCP

De instructie DCP (Decrement and Compare Accumulator) verlaagt de waarde van het gespecificeerde adres met 1 en vergelijkt de uitkomst met de waarde van de accumulator. De waarde van de accumulator blijft hierbij onveranderd.
De Zero-, Negative- en Carry-flag worden net zoals bij een gewone CMP instructie beinvloed.
Adressering: Zero-page; Zero-page,X; Absolute; Absolute,X; Absolute,Y; (Indirect,X); (Indirect),Y

ISC

De instructie ISC (Increment and Subtract with Carry) verhoogt de waarde van het gespecificeerde adres met 1 en trekt de uitkomst met de Carry-flag af van de accumulator.

De Zero-, Negative-, Carry- en overflow flag worden net zoals bij een gewone SBC instructie beinvloed.
Adressering: Zero-page; Zero-page,X; Absolute; Absolute,X; Absolute,Y; (Indirect,X); (Indirect),Y

LAX

De instructie LAX (Load Accumulator and X-register) laadt de waarde van het gespecificeerde adres in zowel de accumulator als in het X-register.
Als deze waarde nul is, wordt de Zero-flag gezet. Als bit 7 van deze waarde aan staat, wordt de Negative-flag gezet.
Adressering: Zero-page; Zero-page,Y; Absolute; Absolute,Y; (Indirect,X); (Indirect),Y

RLA

De instructie RLA (Rotate Left and And accumulator) roteert de inhoud van het gespecificeerde adres een bit naar links. De Carry-flag wordt hierbij in bit 0 van het adres geschoven en bit 7 van het adres wordt in de Carry-flag geschoven. Vervolgens wordt er een logische AND op de uitkomst en de waarde van de accumulator uitgevoerd. De uitkomst van deze AND wordt in de accumulator geplaatst.
Deze instructie beinvloedt de Zero- en de Negative-flag afhankelijk van de uitkomst van de AND. De Carry-flag wordt door de rotatie beinvloed.
Adressering: Zero-page; Zero-page,X; Absolute; Absolute,X; Absolute,Y; (Indirect,X); (Indirect),Y

RRA

De instructie RRA (Rotate Right and Add with carry) roteert de inhoud van het gespecificeerde adres een bit naar rechts. De Carry-flag wordt hierbij in bit 7 van het adres geschoven en bit 0 van het adres wordt in de Carry-flag geschoven. Tenslotte wordt de uitkomst met de Carry-flag bij de accumulator opgeteld.
De Zero-flag wordt gezet als de accumulator nul is. Als bit 7 van de accumulator aan staat, wordt de Negative-flag gezet. De Overflow-flag wordt gezet als bij de optelling de waarde 127 wordt overschreden en de Carry-flag als de waarde 255 wordt overschreden.
Adressering: Zero-page; Zero-page,X; Absolute; Absolute,X; Absolute,Y; (Indirect,X); (Indirect),Y

SLO

De instructie SLO (Shift Left and Or accumulator) schuift de inhoud van het gespecificeerde adres een bit naar links. Bit 7 wordt hierbij in de Carry-flag geschoven en in bit 0 wordt een nul geschoven. Vervolgens wordt er een logische OR uitgevoerd op de uitkomst en de waarde van de accumulator. Het resultaat van deze OR wordt in de accumulator gezet.
De Zero-, Negative- en Carry-flag worden net zoals bij de instructie RLA beinvloed.
Adressering: Zero-page; Zero-page,X; Absolute; Absolute,X; Absolute,Y; (Indirect,X); (Indirect),Y

SRE

De instructie SRE (Shift Right and Exclusive Or accu) schuift de inhoud van het gespecificeerde adres een bit naar rechts. Bit 0 wordt hierbij in de Carry-flag geschoven en in bit 7 wordt een 0 geschoven. Daarna wordt er een logische Exlusive OR uitgevoerd op de uitkomst en de waarde van de accumulator. Het resultaat hiervan wordt in de accumulator opgeslagen.
De Zero-, Negative- en Carry-flag worden weer net zoals bij de instructie RLA beinvloed.
Adressering: Zero-page; Zero-page,X; Absolute; Absolute,X; Absolute,Y; (Indirect,X); (Indirect),Y

Elk van deze acht instructies heeft voor elke adresseringsmode een aparte opcode. Met behulp van figuur 1 is te bepalen wat deze opcode is. Ook is in deze figuur aangegeven welke vlaggen door een instructie beinvloed worden. Een vraagteken betekent dat de vlag beinvloed wordt.

We weten nu al van een flink aantal secret opcodes wat de werking is. Er blijven echter nog een aantal opcodes over die niet in de standaard instructie set van de 6510 voorkomen en ook niet in figuur 1 staan. De meeste van deze opcodes doen helemaal niets. Het zijn zogenaamde NOP instructies (No OPeration). Er zijn drie soorten NOP instructies: een-byte, twee-byte en drie-byte NOP instructies. In de standaard instructie set van de 6510 bevindt zich al een NOP instructie. Dit is een een-byte NOP. Een aantal van de opcodes zijn equivalent met deze NOP. Als de 6510 zo'n NOP tegenkomt, gebeurt er niets en wordt er verder gegaan met de opcode in het adres volgend op het adres van de NOP.
Bij een twee-byte NOP gebeurt er ook niets. Nu wordt echter het adres volgend op de NOP overgeslagen en wordt er verder gegaan met de opcode in het daarop volgende adres.
Bij een drie-byte NOP worden er twee bytes overgeslagen.
In figuur 2 staan de codes voor de verschillende NOP instructies.
Er zijn nu nog maar een paar opcodes over die nog steeds niet beschreven zijn. Deze opcodes laten de 6510 crashen. Als de 6510 een van deze opcodes tegenkomt bij het executeren van een programma, dan is er geen redding meer. Onder luid gevloek een paar keer hard rammen op de STOP/RESTORE toetsen heeft geen zin meer. Het enige dat u nog rest is de resetknop. De opcodes die de 6510 laten crashen, staan ook in figuur 2 vermeld.

Gebruik van de secret opcodes

De mnemonics zoals AAX en DCP kunnen natuurlijk niet met behulp van een gewone assembler of monitor worden ingevoerd. Deze kent immers alleen de mnemonics voor de instructies die in de instructie set van de 6510 beschreven staan. Om de secret opcodes te gebruiken moeten de opcodes direct in het geheugen geplaatst worden. Met een monitor kan dit via het 'M'-commando. In een assembler kan dit gedaan worden met behulp van de '.byt'-pseudo instructie. Achter deze pseudo instructie kan dan als commentaar de mnemonische weergave van de instructie worden opgenomen, zodat het programma makkelijk leesbaar blijft. We zullen dit zo met een voorbeeld programma verduidelijken.
Waarom zou u de secret opcodes gebruiken? Alles wat u met deze opcodes kunt programmeren, kunt u immers ook met de conventionele instructies maken, zonder secret opcodes. Een eerste voordeel is dat het programma korter wordt door het gebruik van secret opcodes. Omdat machinetaal programmeurs zich vaak in de raarste bochten moeten wringen om een programma in de krappe geheugenruimte van de Commodore 64 te persen, kan dit een groot voordeel zijn. Een tweede voordeel is dat het gebruik van secret opcodes programma's soms behoorlijk kan versnellen.
Het volgende programma illustreert het gebruik van secret opcodes. Het programma kan bijvoorbeeld met PAL of PROFI assembler worden ingevoerd.

OPCODES VOOR NOP INSTRUCTIES

een-byte NOP :	$1A, $3A, $5A, $7A, $DA, $FA
twee-byte NOP:	$04, $14, $34, $44, $54, $64 $74, $D4, $F4, $82, $C2, $E2
drie-byte NOP:	$0C, $1C, $3C, $5C, $7C, $DC $FC
crash codes :	$02, $12, $22, $32, $42, $52 $62, $72, $92, $B2, $D2, $F2

Figuur 2

```
LABELS  MNEMONICS
        lda #$00
        sta $fb
        lda #$04
        sta $fc
        ldy #$00
lus0    lda #$20
lus1    .byt $d3,$fb:;dcp ($fb),y
        bne lus1
        iny
        bne lus1
        lda #$08
        .byt $e7,$fc:;isc $fc
        bne lus0
        rts
```

Het programma wist het beeldscherm op een leuke manier. De inhoud van elk scherm adres wordt steeds met een verlaagd totdat de inhoud gelijk is aan 32 (de ascii waarde voor een spatie).
De lengte van dit programma bedraagt 26 bytes. De uitvoeringssnelheid van het programma is ongeveer 3 seconden. (De uitvoeringssnelheid is gemeten bij het wissen van een leeg scherm. Het programma moet dan het meeste werk verrichten.)
Het volgende programma bestaat geheel uit conventionele opcodes. Dit programma is equivalent met het vorige programma.

```
LABELS MNEMONICS
        lda #$00
        sta $fb
        lda #$04
        sta $fc
        ldy #$00
lus1    sec
        lda ($fb),y
        sbc #$01
        sta ($fb),y
        cmp #$20
        bne lus1
        iny
        bne lus1
        inc $fc
        lda $fc
        cmp #$08
        bne lus1
        rts
```

Dit programma is 33 bytes lang en heeft een uitvoeringssnelheid van ongeveer 5 seconden. De twee eerder genoemde voordelen komen hier duidelijk naar voren. Het gebruik van secret opcodes versnelt het programma bijna twee maal. De grootste snelheidswinst wordt door het gebruik van de DCP instructie gehaald. In het tweede programma zijn voor deze ene instructie vijf andere instructie in de plaats gekomen. Dit komt omdat de DEC instructie de adresseringsmode (Indirect),Y niet kent, waardoor de SBC instructie gebruikt moet worden. De SBC instructie werkt alleen op de accumulator, zodat de waarde van het te verlagen adres eerst in de accumulator moet worden geladen en na verlaging weer in het adres moet worden teruggestopt. Tenslotte moet de waarde nog apart met de CMP instructie vergeleken worden. De grote snelheidswinst ontstaat, omdat deze instructies in een lus staan, die 256.000 keer wordt doorlopen (bij het wissen van een leeg scherm). Verder zijn er in het eerste programma twee instructies (LDA en ISC) nodig om de inhoud van adres $FC te verhogen en te vergelijken met de waarde $08. In het tweede programma zijn hiervoor drie instructies nodig. Het tweede programma moet dus ruim een miljoen instructies meer uitvoeren dan het eerste programma.

Secret opcodes lenen zich ook zeer goed voor programmabeveiliging. Door veel secret opcodes te gebruiken wordt een programma onleesbaar als getracht wordt om het programma met een monitor te disassembleren. De monitor drukt namelijk bij de secret opcodes niet de mnemonic van de corresponderende instructie af, maar drie vraagtekens. Vooral de twee- en drie-byte NOP instructies zijn hiervoor geschikt. Als bijvoorbeeld de twee bytes volgend op een drie-byte NOP de waarden voor de instuctie LDA #$00 bevatten, dan drukt de monitor bij het disassembleren het volgende af:

???
LDA #$00

Het lijkt nu net of het programma de instructie LDA #$00 bevat. Deze instructie wordt echter nooit uitgevoerd, omdat de twee bytes volgend op de drie-byte NOP worden overgeslagen. Voor buitenstaanders is het dan vrijwel onmogelijk om het programma te lezen.

Het REM Statement

Met de Basic instructie REM kan commentaar bij een programma worden gezet. Hierbij staat REM voor REMark (opmerking). Na de instructie REM kan een willekeurige regel commentaar ingetypt worden. Als er in een regel een REM statement voorkomt, wordt de rest van de regel (alles na REM) opgevat als commentaar. Er kunnen dus geen andere instructies meer na de REM worden gezet.
Na de REM instructie kunnen alleen ongeshifte letters en cijfers gebruikt worden in het commentaar. Met ongeshifte tekens worden tekens bedoeld, die niet verkregen zijn met de SHIFT toets. U kunt na de REM wel tekens intypen met de shift, maar de computer laat ze dan niet staan. Typt u maar eens het volgende in:

10 rem {shift a} {shift q}

Als u dit programma bekijkt met LIST, ziet u het volgende:

10 rem atn input

U ziet dat de computer van {shift a} atn heeft gemaakt, en van {shift q} input heeft gemaakt. Dit komt omdat de Commodore 64 geshifte tekens na REM als Basic opdrachten interpreteert. Elke Basic opdracht correspondeert met een getal. Zo heeft de PRINT instructie de code 153. ATN heeft dezelfde code als de ascii waarde van {shift a}, en omdat de computer geshifte tekens na REM interpreteert als Basic instructies, wordt {shift a} ATN.
We gaan nu een leuke truc uithalen met REM. De ascii waarde van {shift l} correspondeert niet met een Basic instructie. Met andere woorden, er is geen Basic instructie die als code de ascii waarde van {shift l} heeft. Als we nu {shift l} intypen na de REM, zal de computer tevergeefs de corresponderende instructie proberen te zoeken bij {shift l}. Bij LIST zal dit leiden tot een syntax error. Bij het uitvoeren van het pro-

gramma met RUN, zult u er geen last van hebben, omdat alles na de REM genegeerd wordt. Dit kan dienen als een soort LIST beveiliging. Door bovenaan het programma een regel te zetten, met {shift l}, zal de LIST daar afbreken met een error. De rest van het programma zal dan niet meer gelist worden:

10 rem listing stopt hier {shift l}
20 .. vervolg programma ..

Het is natuurlijk geen echte beveiliging, omdat nu nog gewoon het programma verder gelist kan worden met LIST 20-

Er kunnen nog meer leuke dingen gedaan worden met REM. We hebben al gezegd dat geshifte tekens worden geinterpreteerd als Basic instructies. Verder worden alle ascii codes kleiner dan 128 gewoon afgedrukt op het scherm. Dit lijkt een triviale zin, maar deze houdt ook in, dat er in het REM statement bijvoorbeeld een CLR/HOME karakter kan staan. Bij het listen zal dan het scherm worden schoon gemaakt. Ook kunnen we bijvoorbeeld tijdens het listen de karakters van kleur laten veranderen, door de goede ascii waarde in de REM te zetten. Bij het runnen van het programma hebben we geen last van deze specifieke ascii codes.

De vraag is nu nog, hoe we deze karakters in de REM krijgen, want als we gewoon na REM de CLR/HOME toets indrukken, zal de cursor gewoon naar linksboven springen. Typ nu eens het volgende in:

10 rem""{inst/del}{rvs on}ttttttt{rvs off}helleu there: 6*9 = 42

Een kleine toelichting: na rem moeten eerst twee aanhalingstekens worden ingetypt, en vervolgens moeten de {inst/del} en {rvs on} toets worden ingetypt. Daarna moeten er 7 t's worden ingetypt. De t's komen dan als het goed is in reverse op het scherm. Daarna kan er nog een tekst worden ingevoerd, met de reverse off. Als u deze regel list, zult u het volgende zien:

helleu there: 6*9 = 42

Dit komt, omdat de computer de reverse t interpreteert als een {INST/DEL}. De {INST/DEL} wordt in de quote mode immers ook als een reverse t op het beeld gezet. (De computer zit in de quote mode, als er een haakje openen is ingetypt). Eerst wordt er '10 rem"' op het scherm gezet. De computer voert daarna 7 keer een {INST/DEL} uit. Hierdoor zijn het regelnummer en de letters REM niet meer te zien.

De bovenstaande truc werkt goed voor het {INST/DEL} karakter. Voor de overige karakters werkt deze truc niet. Het heeft bijvoorbeeld geen effect om de het hartje {shift s} in reverse op het scherm te zetten met als doel het scherm te wissen. Wat wel werkt, is het poken van deze karakters in het REM statement. We laten deze manier zien aan de hand van een voorbeeld. U moet eerst de volgende regel intypen:

10 rem from The ArtVark with love

We kunen nu berekenen dat de spatie na REM opgeslagen is in geheugenadres 2054. Immers, het programma begint op adres 2049. De eerste 2 bytes worden gebruikt om het regelnumer (10) in op te slaan. De volgende 2 bytes worden gebruikt om Het beginadres van de volgende regel (die bestaat in dit geval niet) in op te slaan. Vervolgens wordt adres 2053 gebruikt om de code voor REM in op te slaan. Tenslote bevat adres 2054 de spatie. In dit adres kunnen we nu bijvoorbeeld gewoon de ascii waarde van {CLR/HOME} poken:

poke 2054, 19

Let wel, dit is de ascii waarde van een HOME. Dus de cursor wordt linksboven op het scherm gezet, het scherm zelf wordt niet schoongemaakt. Als we nu LIST invoeren, dan zien we inderdaad dat de ingetypte tekst in regel 10, bovenaan het scherm komt te staan.

We kunnen nu van alles in adres 2054 poken. De volgende poke zorgt er bijvoorbeeld voor dat de letterkleur verandert in de kleur wit:

poke 2054, 5

Door achter REM twee spaties te zetten, kunnen er twee ascii waarden na de REM worden gepoked. De tweede spatie heeft dan uiteraard het adres 2055.

Bij het poken van ascii waarden na de REM moeten we er wel mee rekening houden, dat geshifte tekens als Basic instructies worden geinterpreteerd. Dit betekent dat we geen ascii waarden boven de 127 na de REM moeten poken. Het resultaat zal dan immers zijn, dat de geinterpreteerde Basic instructie op het scherm komt. Zo heeft de CLEAR ascii waarde 128. De CLEAR wordt na de REM niet uitgevoerd, maar geinterpreteerd als END.

Error kanaal uitlezen

Het gebeurt soms dat u een file van disk wilt lezen, die achteraf niet blijkt te bestaan. De computer geeft dan de foutmelding FILE NOT FOUND. Bij andere fouten geeft de computer geen foutmelding. Zo kan het zijn dat er geen disk in de diskdrive zit, of dat u naar een programma probeert weg te schrijven onder een naam die al bestaat. Ook kan het zijn dat u een niet bestaand commando aan de diskdrive geeft. In al deze gevallen zal het rode lampje gaan knipperen. Om te weten wat er fout is gegaan, moet het error kanaal van de diskdrive worden uitgelezen. In Commodore Info no.1 febr/mrt '90 hebben we een programma gegeven waarmee het error kanaal uitgelezen kon worden. Dit programma bevatte een INPUT instructie, en daarom moest het error kannaal in een programma worden uitgelezen. De computer geeft immers een ILLEGAL DIRECT ERROR, als er een INPUT of GET instructie direct wordt ingevoerd.

Om het error kanaal uit te lezen, moest dus steeds een programma ingetypt worden. Dit is vaak onhandig, omdat er vaak al een programma in het geheugen staat, dat niet veranderd of weggegooid mag worden. Wat we dus willen, is het error kanaal uitlezen in de directe mode, zonder dat de computer klaagt met een ILLEGAL DIRECT ERROR. Dit kunnen we doen door de in de directe mode de programma mode te simuleren met POKE 58,1. In adres 57 en 58 wordt door de computer het huidige regelnummer van het programma, dat uitgevoerd wordt, bijgehouden. Als er geen programma loopt, staan deze adressen allebei op 0. Door nu een 1 in adres 58 te poken, denkt de computer dat er een programma loopt, en zal dan niet klagen over een INPUT of GET instructie. Hier volgt de one-liner om het error kanaal uit te lezen:

open1,8,15:poke58,1:fori=1to50 :get#1,a$:printa$; :close(st=64):ifst<>64thennext

Dit was het dan weer voor deze aflevering van Tips & Trucs 64. We hopen dat u voldoende kennis heeft opgedaan voor menig uurtje computer plezier. Mocht u zelf nog leuke ideeen hebben voor Tips & Trucs, stuur deze dan op naar de redactie van Commodore Info.

Tot de volgende Tips & Trucs.

Hylke Sprangers & Michel de Boer

GEOS en printers,

hoe zit dat nou....

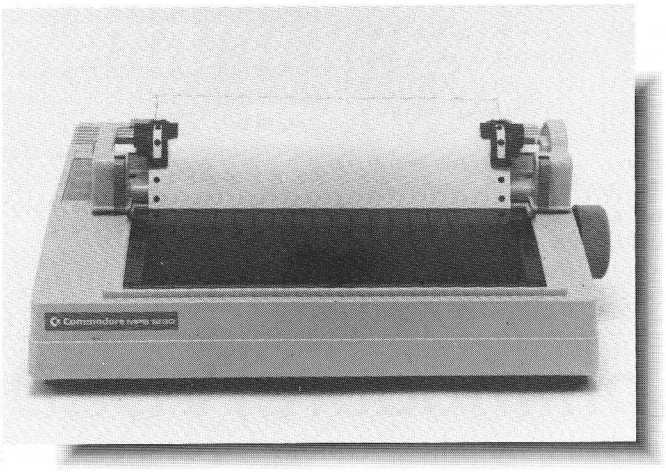
Toen Berkeley in 1986 met GEOS op de markt kwam werden al meer dan 40 printers ondersteund. Maar er kwamen steeds meer printers op de markt die niet alleen voor de Commodore-computers gemaakt werden. Hoe zit dat nou.....

Er zijn in de printerwereld enkele verschillende systemen, formaten eigenlijk. De meest bekende is wel de parallel/centronics-printer.
Deze centronics-standaard is over de hele wereld gelijk en slaat op de pinbelegging van de centronics-stekker.
Daarnaast zijn er de RS-232 seriële printers, dit zijn meestal de printers die aan grote mainframes hangen, o.a. de zgn. regeldrukkers. Uw bankafschrift wordt bijvoorbeeld door zo'n printer gemaakt. En dan hebben we de zogenaamde 'custom made' printers. Dit zijn printers die naar de specificaties van de klant door de fabrikant worden gemaakt. De Commodore printers zijn hier een welsprekend voorbeeld van. De printers kunt u alleen aan een Commodore hangen en niet aan een ander merk computer.

Interface

Nu waren deze Commodore-printers best in staat om hetgeen wat op het scherm stond op papier te zetten maar kwaliteit stond niet hoog in het vaandel. Iedereen heeft wel een uitdraai van een MPS 801 gezien. Om te huilen. Een matrix van 6x7. Een hoge g en een P was niet te vergelijken met een p.

Maar met programma's zoals Koala Painter en Doodle stonden deze printers hun mannetje. Tekstverwerking was eigenlijk een verhaal apart. Mooie brieven konden er niet mee worden gemaakt. Een brief gemaakt met Easy Script dorst je niet naar de belastinginspecteur te sturen. Maar een centronics printer was niet aan te sluiten aan een Commodore. Dus bedacht men een interface die van de seriële poort naar een centronics-aansluiting liep. Een nadeel van deze interfaces was de prijs. Een paar honderd gulden moet je al snel voor deze interfaces neertellen.
Met een interface tussen de C 64/128 en een centronics-printer gaf de kwaliteit waar de printer voor stond. NLQ lag nu ook in het bereik van de Commodore gebruiker. Ook kwamen er steeds meer programma's die een printermenu tot hun beschikking hadden waarin het mogelijk was een centronics-printer te gebruiken.



Printerdrivers

Met de komst van GEOS kwam de printer pas eigenlijk goed in het daglicht te staan. Voor iedere denkbare printer was een apart stukje software geschreven, eigenlijk is er meer sprake van een printer-routine die door de software wordt aangesproken en waarvan enkele parameters worden aangepast aan de printer die wordt gebruikt. Het gevolg is dat er geen printer is waarvoor geen printerdriver kon worden gemaakt.
In eerste instantie waren alle printerdrivers gebaseerd op de Commodore MPS 801-karakterset. Dit is een 60 dpi(dots per inch) formaat. De centronics-printers kregen een eigen driver mee van Berkeley. Maar nog steeds via de seriële poort en via een interface.
Ook werden er hele luxe printerdrivers gemaakt voor bijvoorbeeld de HP laserjet en de Apple LaserWriter. Printers waar een prijskaartje aan hangt van enkele duizenden guldens. Voor de Apple LaserWriter ging men zelfs zover dat de in de printer aanwezige letterFonts automatisch werden gekozen bij gebruik van een bepaald lettertype. U kent de lettertypes wel, u heeft ze vast wel eens gezien: de LW_ROMA, LW_California etc. Nu weet u ook waar de LW voor staat. Voor de LaserWriter.

GeoCable

Voor deze dure printers kwam de hobbyist niet in aanmerking en bleef zitten met de overige drivers. De kwaliteit van de centronics-printers bleef zich bij iedere nieuwe printer verbeteren. NLQ, MultiFonts en 24 naalds werden hele gewone uitspraken. Maar erg ver kwam je er niet mee. Met de MPS 801-standaard kon je niet gebruik maken van al dat mooie wat de centronics-printer zijn eigenaar kon bieden.
Met veel poeha werd door Berkeley een geoCable aangekondigd.
Dit was een kabel die aangesloten kon worden rechtstreeks op de Userpoort en de centronics-printer. De bijbehorende printerdrivers sturen de informatie rechtstreeks naar de Userpoort.
Dus zonder tussenkomst van een dure interface.
Je zou bijna kunnen zeggen dat je een aantal bedrijven zonder werk zou zetten, want met een rechtstreekse aansturing van de userpoort zouden geen interfaces meer nodig zijn.
Of dit effect heeft gehad op de aankondiging weet ik niet maar de geoCable is nooit in produktie genomen door Berkeley.

Kabel zelf maken

Vreemd genoeg verschenen op het Qlink bulletinboard wel de genoemde centronics drivers. Deze drivers vielen op doordat zij de toevoeging (gc) meekregen. Door de centronics-drivers alsnog beschikbaar te stellen ontliep Berkeley de beschuldiging van de interfaceb-edrijven. Men kon met de drivers doen wat men wilde.
Men ging ervan uit dat bij aankoop van de printer een universele printerkabel werd meegeleverd. Berkeley daarentegen had de geoCable al een heel eigen standaard meegegeven en bleek dan achteraf ook niet compatibel met de meeste universele printerkabels. Dit was het resultaat van enig speurwerk in de zogenaamde (gc) drivers. Het enige wat op het scherm kwam te staan was de bekende foutmelding 'Printer is inaccessable'.
Terug naar de tekentafel dus. Wat ons nu nog overbleef was om zelf een kabel te maken. Dat lukte ons inderdaad en het voordeel wat wij hiermee behaalden was dat wij nu zelf ook de lengte van de kabel konden bepalen. De 1.20 meter van de universele printerkabel is toch wel wat aan de korte kant.
Wij gebruiken nu een kabellengte van 2 meter zonder problemen.

Printer Creator

Nu zijn wij zover dat er inmiddels programma's zijn zoals Printer Creator waarmee nieuwe printerdrivers gemaakt kunnen worden.
Of het nu om een seriële of centronics-printer gaat is niet meer belangrijk. Alles wat men nu nog nodig heeft is het boekje met Printer Control Commando's wat met de printer meegeleverd wordt. Hierin vindt u alle opties die uw printer levert, compleet met Escape-codes. Een voorbeeld: Softwarematig NLQ kiezen kan bijvoorbeeld met print chr$(27)+chr$(120)+chr$(49) in Basic.
De Printer Creator daarentegen vraagt in een menu om de volgende invoer: 27,120,49. Dit is al een stuk gebruiksvriendelijker.
Tevens kan men met dit soort programma's bepalen hoeveel keer dezelfde printergang gemaakt wordt. Hierdoor kun je met een grijs lint toch nog een zwarte afdruk realiseren. Deze drivers worden dan ook wel de DS (Double Strike = 2x printen) en QS (Quadruple Strike = 4x printen) genoemd. Ook zijn er gewone 9 naalds printers die door middel van een grafische mode te selecteren een dpi leveren van 240. Men spreekt dan van een Quadruple density (een viervoudige dichtheid). En dit terwijl een Laserprinter een resolutie haalt van 300 dpi. Niet slecht voor een 9 naalds printer dus.

Tot Slot

Een printer aansluiten hoeft nu geen probleem meer te zijn.
Of deze nu serieel of centronics is, GEOS heeft er een driver voor. Mocht u toch onverhoopt met enkele vragen blijven zitten, probeer dan eens contact te zoeken met de St. GEOS Gebruikers.
Zij kunnen u waarschijnlijk alles over deze nieuwe printerdrivers vertellen.

Pim Broekhuijzen

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

De laatste tijd wordt er weer veel gebeld zodat U nogwel eens in gesprek krijgt, daarom houdt uw vraag kort, vermeld in welk blad het desbetreffende artikel stond. Heeft U veel vragen, of zis uw vraag erg uitgebreid, doe het dan schriftelijk, zodat we veel mensen op de maandag avond te woord kunnen staan.

```
  1 rem ******************************
  2 rem basic loader "SYNTAX.CHECKSUM"
  3 rem na de commando's "run" en "new"
  4 rem blijft dit programma in het ge-
  5 rem heugen. laad het te testen pro-
  6 rem gramma en tik daarna sys 49152.
  7 rem ******************************
 10   i=49152  :rem beginadres
 20   reada:ifa<0then40:rem data ingelezen
 30   pokei,a:i=i+1:b=b+a:goto20
 40   if b<>16844thenprint"[SHIFT-CLR]fo
      ut[SPACE]in[SPACE]dataregels!":b=0:end
 50   poke49184,148:poke49185,192
 55   i=49300
 60   read a: ifa<0then80
 70   pokei,a:b=a+b:i=i+1:goto60
 80   if b<>20068thenprint"[SHIFT-CLR]fo
      ut[SPACE]in[SPACE]dataregels![SPAC
      E](vanaf[SPACE]regel[SPACE]240)":b
      =0:end
 90   print"data[SPACE]is[SPACE]weggezet"
 95   print"checksum[SPACE]testen[SPACE]
      met[SPACE]sys49152"
100   data 165,43,166,44,133,163,134,164
      ,169,147
110   data 32,210,255,160,0,240,3,32,73,192
120   data 32,73,192,208,1,96,32,225,255,208
130   data 3,76,116,164,32,81,192,32,73,192
140   data 240,12,201,32,240,247,24,101,
      167,133
150   data 167,76,37,192,166,167,169,0,1
      32,168
160   data 32,205,189,169,13,32,210,255,
      164,168
170   data 76,17,192,200,208,2,230,164,1
      77,163
180   data 96,162,0,189,123,192,240,6,32,210
190   data 255,232,208,245,32,73,192,170
      ,32,73
200   data 192,132,168,32,205,189,162,3,
      169,32
210   data 32,210,255,202,208,250,169,0,
      133,167
220   data 164,168,96,82,69,71,69,76,32,0
230   data  -1
240   data 165,197,201,3,240,7,201,4,240
250   data 6,76,148,192,76,34,192,169
260   data 147,32,210,255,76,161,192
270   data -1
```

** EINDE LISTING checksum 64 **

Checksum Checksum 64

REGEL 1	249	REGEL 120	232
REGEL 2	84	REGEL 130	183
REGEL 3	105	REGEL 140	96
REGEL 4	2	REGEL 150	96
REGEL 5	246	REGEL 160	127
REGEL 6	152	REGEL 170	71
REGEL 7	249	REGEL 180	223
REGEL 10	157	REGEL 190	73
REGEL 20	64	REGEL 200	79
REGEL 30	38	REGEL 210	109
REGEL 40	57	REGEL 220	106
REGEL 50	14	REGEL 230	225
REGEL 55	251	REGEL 240	16
REGEL 60	192	REGEL 250	163
REGEL 70	42	REGEL 260	92
REGEL 80	244	REGEL 270	22
REGEL 90	245		
REGEL 95	237		
REGEL 100	183		
REGEL 110	158		

PRINT OUT C-64 met o.a. GeosIkon

Break

Dit programma werkt niet samen met een ingeschakelde cartridge! De opzet van het spel is erg eenvoudig: alle ballen die je ziet wegschieten. Het lijkt erg eenvoudig, het spel is niet snel, maar schijn bedriegt. Het programma is geschreven door Wim de Wild uit Nigtevecht.

```
10  rem ***************************
20  rem *   don't break it !       *
30  rem *   1989 pitsoft           *
40  rem *   by wim de wild         *
50  rem *   turn out cartridge !! *
60  rem ***************************
70  poke53281,0:poke53280,0:print"[SHI
    FT CLR][CTRL-3]"chr$(8)"[7xSPACE]p
    lease[SPACE]wait[SPACE]!!![10xSPAC
    E]"
80  poke56334,0:poke1,51:fork=0to511:p
    oke14336+k,peek(53248+k):next:poke
    1,55
90  poke56334,1:poke53270,216:dimc(14)
100 sid=54272:fork=0to26:poke54272+k,0
    :next
110 pokesi+24,15:pokesi+5,6:pokesi+6,2
    51:pokesi+4,129
120 fork=0to7:poke15616+k,255:next
130 data21,127,255,255,255,255,63,0
140 data82,252,252,252,252,252,248,0
150 data15,11,12,10
160 data14,11,14,10
170 data10,15,11,12
180 data15,14
190 fork=0to15:reada:poke14848+k,a:nex
    t
200 fork=0to13:readc(k):next:poke53282
    ,1:gosub1000:gosub1340:poke53272,3
    0
210 print"[SHIFT-CLR]":forl=1to8:poke6
    46,c(rnd(0)*8):gosub930:next:sc=12
    0:ps=0:le=1:f=0
220 li=3:print"[2xCRSR-DOWN][CTRL-2][9
    xSPACE]don't[SPACE]break[SPACE]it"
230 print"[9xSPACE]by[SPACE]wim[SPACE]
    de[SPACE]wild"
240 print"[9xSPACE]1989[SPACE]pitsoft"
250 sys49177:pokev+2,100:pokev+3,200:p
    oke52992,179:poke52993,7:gosub1050
260 poke52994,2:sys49474:print"[HOME]h
    i-score[12xSPACE][11xCRSR-LEFT]";h
    i:pokesi+4,32
270 if(peek(56320)and16)=0then490
280 f=f+1:iff<>300then270
290 sys49487
300 print"[SHIFT-CLR][SPACE]welcome[SP
    ACE]at[4xSPACE]don't[SPACE]break[S
    PACE]it[SPACE]!!!!!"
310 print"[SPACE]created[SPACE]by[SPAC
    E]wim[SPACE]de[SPACE]wild."
320 print"[SPACE]defend[SPACE]your[SPA
    CE]wall[SPACE]against[SPACE]enemie
    s,"
330 print"[SPACE]but[SPACE]do[SPACE]no
    t[SPACE]hit[SPACE]your[SPACE]wall[
    SPACE]when[SPACE]you[SPACE]want"
340 print"[SPACE]to[SPACE]hit[SPACE]th
    e[SPACE]ball."
350 pokev+23,1:pokev+29,1:pokev+16,1:p
    okev+1,100:pokev+21,1
360 fork=356to150step-1:pokev+16,k/256
    :pokev,kand255:next
370 fork=100to227:pokev+1,k:next
380 a$="[18xSPACE][SHIFT-*][21xSPACE]"
390 b$="[40xSPACE]"
400 fork=20to0step-1:poke214,k+1:print
    :printb$;:poke214,k:print:printa$;
    :next
410 print"[SHIFT-CLR]your[SPACE]oppone
    nt[SPACE]cannot[SPACE]be[SPACE]qui
    et[SPACE]for[SPACE]a"
420 print"moment.don't[SPACE]touch[SPA
    CE]him[SPACE]or[SPACE]you[SPACE]ar
    e[SPACE]gone!"
440 fork=1to3000:next:pokev+21,2:pokev
    +40,2:pokev+28,3:pokev+23,0:pokev+
    29,0
450 pokev+3,58
460 fork=336to25step-8:poke2041,193
470 x=(k-24)/8:print"[HOME]"tab(x)"[SP
    ACE][CRSR-DOWN][CRSR-LEFT][SPACE][
    CRSR-DOWN][CRSR-LEFT][SPACE][CRSR-
    DOWN][CRSR-LEFT][SPACE][CRSR-DOWN]
    [CRSR-LEFT][SPACE][CRSR-DOWN][CRSR
    -LEFT][SPACE][CRSR-DOWN][CRSR-LEFT
    ][SPACE]":pokev+16,int(k/256)*2
480 pokev+2,kand255:next:pokev+21,1:go
    to210
490 sys49487:print"[SHIFT-CLR]":gosub9
    50:pokev+21,3:x=int(rnd(0)*264)+24
    :pokev+2,xand255
500 pokev+16,int(x/256)*2:raak=0
510 m=leand7:ds=le*70:dr=int(le/8)*4+4
    :ifm>=7thenm=0
520 n=((7-m)and7):ifn=0thenn=2
530 forl=0ton:poke646,c(2*m+(land1)):g
    osub930:next:poke52992,59+l*16
540 k=int(le/8)+1:poke52993,k:pokev+21
    ,2[KWADRAAT PIJL](k+1)-1:m=leand7:
    ifm>=5thenm=4550poke52994,m:sy
    s49474
560 sys49474:pokesi+4,128
570 geta$:ifa$="[PIJL LINKS]"then210
580 ifa$="p"thensys49487:wait56320,16,
    16:fork=0to500:next:sys49474
590 j=peek(56320)and16:gosub850:ifxxth
    en880
600 ifjthen570
610 pokesi+4,129
620 sys49487:m=peek(v)+256*(peek(v+16)
    and1):x=int((m-16)/8):y=22:b=1024+
    y*40+x
630 pokeb+40,32:fr=rnd(0)*900+256:poke
    si+1,fr/256:pokesi,frand255
640 pokeb,9:pokeb+54272,(rnd(0)*7)+8:g
    osub850:ifxxthen880
650 pokev+31,0:fork=0to50:next:ifpeek(
    v+31)=0theny=y-1:goto730
660 pokev+28,1:pokesi+1,5:pokesi+4,32:
    fork=0to1000:next:pokev+28,255
670 pokeb,32:d=(24-y)*40:sc=sc+d:ps=ps
    +d:gosub950:ra=ra+1:ifra<>drthen560
680 ra=0:le=le+1:ifle<>55then490
690 print"[SHIFT-CLR]":sys49487:printt
    ab(15)"[CTRL-5]congrats[SPACE]!!!!
    !![SPACE]you[SPACE]survived[SPACE]
```

```
     the[SPACE]ball-war"
700  print"[CTRL-3][CRSR-DOWN]your[SPAC
     E]optimal[SPACE]score[SPACE]is[SPA
     CE]:"ps
710  print"your[SPACE]minimal[SPACE]sco
     re[SPACE](including[SPACE]negative
     [SPACE]scores)[SPACE]:"sc
720  print"[CTRL-8][2xCRSR-DOWN]"tab(10
     )"press[SPACE]fire[SPACE]to[SPACE]
     continue.":wait56320,16,16:sys4947
     4:goto210
730  l=int(x/2)*2:c=(b-x)+l-40:ifpeek(c
     )=32then770
740  pokec,32:pokec+1,32:pokeb,32:sc=sc
     -ds:pokesi,0:pokesi+1,3:pokesi+4,3
     2
750  fork=0to500:next:ifsc<0thenli=0:go
     to880
760  gosub950:goto560
770  ify=1thenpoke53280,7:fork=0to100:n
     ext:sc=sc-400:poke53280,0:goto560
780  b=b-40:goto630
790  sys49487:y=24:m=peek(v)+256*(peek(
     v+16)and1):l=peek(v+28):pokev+28,l
     -1
800  fork=0to2:x(k)=int(m/8)-2:next:pok
     esi+4,129:pokesi+1,1:pokesi,120
810  fork=0to2:a=1024+y*40+x(k):b=a+542
     72:pokea,160:pokeb,rnd(0)*6+1
820  next:a=a-x(2):fork=0to2:pokea+x(k)
     ,32:next
830  x(0)=x(0)+1:x(2)=x(2)-1:y=y-1:ify<
     >20then810
840  sys49474:pokev+28,l:pokesi+4,32:re
     turn
850  pokev+30,0:if(peek(v+30)and1)=1the
     nli=li-1:gosub950:gosub790
860  xx=0:ifli<>0thenreturn
870  xx=1:return
880  sys49487:print"[SHIFT-CLR][CTRL-7]
     [SPACE]see[SPACE]you[SPACE]later[S
     PACE]alligator,it's[SPACE]game[SPA
     CE]over"
890  pokesi+4,32
900  print"[2xCRSR-DOWN]"tab(13)"[CTRL
     8]score:[CTRL-4]"ps:printtab(13)"[
     CTRL-3]press[SPACE]fire!":wait5632
     0,16,16
910  ifps>hithenhi=ps:goto210
920  goto210
930  fork=0to39:print"[SHIFT-*]A";:next
940  return
950  print"[HOME][39xSPACE]";
960  print"[HOME][CTRL-2]score:"sc"[SPA
     CE]level:"le"[SPACE]ships:"li"[3xS
     PACE][3xCRSR-LEFT]":return
970  data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0,0,0,0,24,0,0,24,0,0,
     24,0
980  data0,24,0,0,24,0,0,24,0,0,126,0,1
     ,255,128,129,255,130,131,235,194,1
     79,190
990  data206,191,235,254,255,255,255,0
1000 fork=0to127:reada:poke12288+k,a:ne
     xt
1010 poke2040,192:v=53248:pokev,168:pok
     ev+1,227
1020 pokev+23,0:pokev+29,0
1030 t=0:fork=0to351:reada:t=t+a:poke49
     152+k,a:next:ift=46209thenreturn
1040 print"data[SPACE]error!":end
1050 pokev+39,12:pokev+38,11:pokev+37,1
     :fork=0to7:pokev+40+k,2+k:next
1060 fork=1to7:poke2040+k,193:pokev+2*k
     ,int(rnd(0)*200)+24
1070 pokev+2*k+1,int(rnd(0)*40)+179:nex
     t:pokev+21,255:pokev+28,255:return
1080 data0,175,0,2,170,192,10,170,176,1
     0,170,176,41,170,172,41
1090 data170,172,42,170,172,42,170,172,
     42,170,168,42,170,168,42,170
1100 data168,10,170,160,10,170,160,2,17
     0,128,0,170,0,0,0,0
1110 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0
1120 data168,169,0,141,255,207,56,46,25
     5,207,24,136,16,249,96,165
1130 data2,10,133,251,169,208,133,252,9
     6,162,7,169,1,157,240,206
1140 data157,224,206,202,16,247,96,32,1
     5,192,165,2,32,0,192,164
1150 data2,169,0,153,224,206,168,177,25
     1,201,63,208,16,173,16,208
1160 data45,255,207,240,8,164,2,169,1,1
     53,224,206,96,24,177,251
1170 data105,1,145,251,208,246,173,16,2
     08,13,255,207,141,16,208,96
1180 data32,15,192,165,2,32,0,192,164,2
     ,169,1,153,224,206,160
1190 data0,173,16,208,45,255,207,208,14
     ,177,251,201,24,208,8,164
1200 data2,169,0,153,224,206,96,56,177,
     251,233,1,145,251,201,255
1210 data208,244,173,255,207,73,255,141
     ,255,207,173,16,208,45,255,207
1220 data141,16,208,96,169,0,133,2,173,
     0,220,41,15,201,7,240
1230 data7,201,11,208,238,76,96,192,76,
     39,192,32,15,192,230,251
1240 data164,2,185,240,206,201,1,240,17
     ,160,0,177,251,205,0,207
1250 data208,24,169,1,164,2,153,240,206
     ,96,160,0,177,251,201,228
1260 data208,8,169,255,164,2,153,240,20
     6,96,160,0,177,251,164,2
1270 data24,121,240,206,160,0,145,251,9
     6,173,1,207,141,3,207,173
1280 data2,207,141,4,207,172,3,207,132,
     2,185,224,206,201,1,240
1290 data6,32,39,192,76,26,193,32,96,19
     2,32,187,192,206,4,207
1300 data173,4,207,16,224,173,2,207,141
     ,4,207,206,3,207,173,3
1310 data207,208,210,173,1,207,141,3,20
     7,32,164,192,234,234,234,76
1320 data49,234,120,169,192,141,21,3,16
     9,249,141,20,3,88,96,120
1330 data169,49,141,20,3,169,234,141,21
     ,3,88,96,0,0,0,0
1340 print"[SHIFT-CLR][CTRL-3][CTRL-N]W
     elcome[SPACE]at[SPACE][CTRL-2]Don'
     t[SPACE]break[SPACE]it[SPACE]!"
1350 print"[CTRL-6]This[SPACE]is[SPACE]
     a[SPACE]game[SPACE]which[SPACE]you
     [SPACE]can[SPACE]play[SPACE]simp-"
1360 print"ly.Just[SPACE]shoot[SPACE]do
```

```
     wn[SPACE]all[SPACE]balls[SPACE]you
     [SPACE]see."
1370 print"You[SPACE]better[SPACE]don't
     [SPACE]miss,because[SPACE]that[SPA
     CE]will"
1380 print"give[SPACE]you[SPACE]a[SPACE
     ]giant[SPACE]repairbill[SPACE]for[
     SPACE]your"
1390 print"score.If[SPACE]your[SPACE]sc
     ore[SPACE]is[SPACE]negative,it[SPA
     CE]is"
1400 print"simply[SPACE]game[SPACE]over
     .Remember:When[SPACE]a[SPACE]ball"
1410 print"is[SPACE]close[SPACE]to[SPAC
     E]a[SPACE]wall[SPACE]and[SPACE]you
     [SPACE]hit[SPACE]it,you'll";
1420 print"get[SPACE]more[SPACE]bonus-p
     oints[SPACE]than[SPACE]a[SPACE]bal
     l[SPACE]which"
1430 print"is[SPACE]far[SPACE]from[SPAC
     E]the[SPACE]wall[SPACE](=close[SPA
     CE]to[SPACE]you)."
1440 print"You'll[SPACE]loose[SPACE]one
     [SPACE]of[SPACE]your[SPACE]three[S
     PACE]lives"
1450 print"when[SPACE]the[SPACE]ball[SP
     ACE]hits[SPACE]you[SPACE]badly[2xS
     PACE](not"
1460 print"always[SPACE]!!).[SPACE]Ther
     e[SPACE]is[SPACE]a[SPACE]penalty[S
     PACE]of[SPACE]400"
1470 print"if[SPACE]you[SPACE]break[SPA
     CE]a[SPACE]wall.Use[SPACE][PIJL LI
     NKS][SPACE]for[SPACE]exit[SPACE]and"
1480 print"'p'[SPACE]for[SPACE]pause.Pr
     ess[SPACE]the[SPACE]'firebutton'[S
     PACE]"
1490 print"or[SPACE]'p'[SPACE]to[SPACE]
     continue.This[SPACE]game[SPACE]is"
1500 print"joystick[SPACE]required[SPAC
     E]in[SPACE]port[SPACE]2[SPACE](the
     [SPACE]one"
1510 print"nearest[SPACE]the[SPACE]powe
     r-switcher)."
1520 print"[CTRL-8][5xSPACE]Press[SPACE
     ]fire[SPACE]to[SPACE]continue"
1530 wait56320,16,16:wait56320,127
1540 print"[SHIFT-CLR]Design,Graphics,S
     ound[SPACE]and[SPACE]Programming[S
     PACE]:"
1550 print"Wim[SPACE]de[SPACE]Wild[SPAC
     E],[SPACE]Oostzijdestraat[SPACE]62"
1560 print"1393[SPACE]pn[SPACE]in[SPACE
     ]Nigtevecht[SPACE](Netherlands)
1570 print"[CRSR-DOWN][CTRL-2]Any[SPACE
     ]questions[SPACE]?[2xSPACE]Just[SP
     ACE]ring[SPACE]me[SPACE]:"
1580 print"[8xSPACE][CTRL-7]02945-1984"
1590 print"[2xCRSR-DOWN][CTRL-2]Press[S
     PACE]fire.If[SPACE]you[SPACE]press
     [SPACE]fire[SPACE]again"
1600 print"this[SPACE]game[SPACE]will[S
     PACE]start."
1610 wait56320,16,16:return
```

** EINDE LISTING break

Checksum break

REGEL 10	253
REGEL 20	98
REGEL 30	231
REGEL 40	36
REGEL 50	251
REGEL 60	253
REGEL 70	209
REGEL 80	4
REGEL 90	73
REGEL 100	108
REGEL 110	167
REGEL 120	214
REGEL 130	189
REGEL 140	236
REGEL 150	147
REGEL 160	148
REGEL 170	147
REGEL 180	122
REGEL 190	178
REGEL 200	190
REGEL 210	138
REGEL 220	22
REGEL 230	30
REGEL 240	225
REGEL 250	162
REGEL 260	177
REGEL 270	43
REGEL 280	91
REGEL 290	174
REGEL 300	20
REGEL 310	68
REGEL 320	59
REGEL 330	25
REGEL 340	143
REGEL 350	215
REGEL 360	121
REGEL 370	131
REGEL 380	27
REGEL 390	92
REGEL 400	27
REGEL 410	44
REGEL 420	88
REGEL 440	187
REGEL 450	99
REGEL 460	18
REGEL 470	31
REGEL 480	46
REGEL 490	0
REGEL 500	235
REGEL 510	219
REGEL 520	133
REGEL 530	194
REGEL 540	111
REGEL 550	1
REGEL 560	188
REGEL 570	191
REGEL 580	62
REGEL 590	69
REGEL 600	24
REGEL 610	217
REGEL 620	128
REGEL 630	150
REGEL 640	157
REGEL 650	102
REGEL 660	239
REGEL 670	113
REGEL 680	198
REGEL 690	236
REGEL 700	100
REGEL 710	143
REGEL 720	35
REGEL 730	90
REGEL 740	53
REGEL 750	98
REGEL 760	137
REGEL 770	39
REGEL 780	161
REGEL 790	221
REGEL 800	25
REGEL 810	83
REGEL 820	49
REGEL 830	91
REGEL 840	1
REGEL 850	208
REGEL 860	181
REGEL 870	91
REGEL 880	1
REGEL 890	162
REGEL 900	50
REGEL 910	83
REGEL 920	28
REGEL 930	77
REGEL 940	142
REGEL 950	43
REGEL 960	69
REGEL 970	213
REGEL 980	210
REGEL 990	35
REGEL 1000	226
REGEL 1010	40
REGEL 1020	240
REGEL 1030	213
REGEL 1040	92
REGEL 1050	240
REGEL 1060	149
REGEL 1070	231
REGEL 1080	164
REGEL 1090	177
REGEL 1100	195
REGEL 1110	23
REGEL 1120	108
REGEL 1130	30
REGEL 1140	236
REGEL 1150	81
REGEL 1160	246
REGEL 1170	129
REGEL 1180	123
REGEL 1190	29
REGEL 1200	29
REGEL 1210	38
REGEL 1220	64
REGEL 1230	251
REGEL 1240	208
REGEL 1250	24
REGEL 1260	243
REGEL 1270	9
REGEL 1280	155
REGEL 1290	162
REGEL 1300	121
REGEL 1310	120
REGEL 1320	244
REGEL 1330	83
REGEL 1340	191
REGEL 1350	136
REGEL 1360	138
REGEL 1370	69
REGEL 1380	213
REGEL 1390	181
REGEL 1400	86
REGEL 1410	120
REGEL 1420	123
REGEL 1430	105
REGEL 1440	92
REGEL 1450	197
REGEL 1460	98
REGEL 1470	45
REGEL 1480	76
REGEL 1490	192
REGEL 1500	167
REGEL 1510	74
REGEL 1520	182
REGEL 1530	74
REGEL 1540	172
REGEL 1550	33
REGEL 1560	58
REGEL 1570	39
REGEL 1580	3
REGEL 1590	132
REGEL 1600	35
REGEL 1610	128

GeosIkon

Ook van Wim de Wild is het programma waarmee Geos-ikonen kunnen worden aangepast. Het eigenlijke programma bestaat uit twee delen, het ophalen van een ikoon, en het gedeelte waarbij deze ikoon kan worden bewerkt. Het programma werkt alleen als u ook de Geos diskette erbij gebruikt. Een verdere uitleg is in het programma opgenomen.

```
10  rem *****  geos icon editor   *****
20  rem *****  ----------------   *****
30  rem *****                     *****
40  rem *****    1989 pitsoft     *****
50  rem *****    wim de wild      *****
60  goto710
70  print"put[SPACE]a[SPACE]geos[SPACE
    ]disk[SPACE]in[SPACE]drive:a[SPACE
    ]and[SPACE]press[4xSPACE]any[SPACE
    ]key.":wait198,1:poke198,0
80  fork=0to63:poke832+k,0:next
90  v=53248:pokev+21,1:poke2040,13:pok
    ev,20:pokev+1,211:print"[SHIFT-CLR
    ]":pokev+33,12
100 pokev+32,0:pokev+39,11:dimtc(50),s
    c(50):pokev+16,1
101 rem ******** grabbing *********
110 s=1:i=1:cs=s
120 open1,8,15:open2,8,2,"#"
130 print#1,"u1";2;0;18;cs:get#2,t$,s$
    :s=asc(s$+chr$(0)):a=21
140 fork=1to8:print#1,"u1";2;0;18;cs:p
    rint#1,"b-p[SPACE]2";a:get#2,t$
150 tc(i)=asc(t$+chr$(0)):get#2,s$:sc(
    i)=asc(s$+chr$(0)):a=a+32:i=i+1:nextk
160 cs=s:ifs<>255then130
170 fork=1toi-1
180 iftc(k)=0thennext
190 print#1,"u1";2;0;tc(k);sc(k):print
    #1,"b-p[SPACE]2[SPACE]5":forl=1to6
    3:get#2,r$
200 poke831+l,asc(r$+chr$(0)):next
210 print"[SHIFT-CLR][COM-4]1)[SPACE]e
    dit":print"2)[SPACE]copy[SPACE]to[
    SPACE]buffer":print"3)[SPACE]next"
    :print"4)[SPACE]previous"
220 geta$:ifa$=""then220
230 n=val(a$):ifn<1orn>4then220
240 onngoto290,250,280,260
250 forl=0to63:poke704+l,peek(832+l):n
    ext:goto210
260 k=k-1:ifk=0thenk=1
270 goto180
280 next:goto170
290 tr=tc(k):se=sc(k):close2:close1
291 rem ********* editing *********
300 print"[SHIFT-CLR]";:d$="[CTRL-9][C
    OM 4][24xSPACE]":fork=0to20:printd
    $:next
310 poke214,22:print:print"[COM-4][4xS
    PACE]wim[SPACE]de[SPACE]wild[SPACE
    ]1989[SPACE]!!![HOME]";
320 print"[COM-4][HOME]"tab(25)"edit:"
330 printtab(25)"1[SPACE]inverse"
340 printtab(25)"2[SPACE]write"
350 printtab(25)"3[SPACE]grabbing"
360 printtab(25)"4[SPACE]geos"
370 printtab(25)"5[SPACE]basic[SPACE]2.0"
380 printtab(25)"6[SPACE]buff-screen":
    printtab(25)"7[SPACE]clear[SPACE]s
    creen"
390 printtab(25)"[space][SPACE]dot-"
400 printtab(25)"[return][SPACE]dot+"
410 fory=0to20:forx=0to23step8
420 b=55296+y*40+x:m=peek(832+y*3+int(
    x/8)):forn=7to0step-1
430 f=sgn(mand2[KWADRAAT PIJL]n):p=12:
    p=p+(f=1):poke b+(7-n),p:nextn,x,y
440 y=0:x=0:r$=chr$(13)
450 co=(55296+y*40+x):mem=peek(co):pokeco,
460 geta$:ifa$=""then460
470 ifa$="[CRSR-DOWN]"thenify<20theny=y+1
480 ifa$="[CRSR-UP]"thenify>0theny=y-1
490 ifa$="[CRSR-RIGHT]"thenifx<23thenx=x+1
500 ifa$="[CRSR-LEFT]"thenifx>0thenx=x-1
510 f=832+int(x/8)+y*3:ifa$=r$thenmem=
    11:pokef,peek(f)or2[KWADRAAT PIJL]
    (7-(xand7))
520 ifa$="[SPACE]"thenmem=12:l=255-(2[
    KWADRAAT PIJL](7-(xand7))):pokef,p
    eek(f)andl
530 pokeco,mem
540 ifa$="7"thenforz=0to63:poke832+z,0
    :next:print"[HOME]";:forz=0to20:pr
    intd$:next
550 ifa$="1"thenforz=0to63:poke832+z,2
    55-peek(832+z):next:goto410
560 ifa$="2"then630
570 ifa$="3"then110
580 ifa$="4"then670
590 ifa$="5"thensys64738
600 ifa$="6"then620
610 goto450
611 rem ******** buffer option ********
620 forl=0to63:poke832+l,peek(704+l):n
    ext:goto410
621 rem ********* write option ********
630 open3,8,15:open4,8,2,"#":print#3,"
    b-p:"2;5
640 forz=1to63:print#4,chr$(peek(831+z
    ));:next
650 print#3,"b-w:"2;0;tr;se
660 close4:close3:run
670 poke53281,6:poke53280,14:print"[CO
    M 7][SHIFT-CLR]Doe[SPACE]de[SPACE]
    Bootdisk[SPACE]in[SPACE]Drive:A"
680 print"en[SPACE]druk[SPACE]een[SPAC
    E]toets."
690 wait198,1:print"[SHIFT-CLR][CTRL-7
    ]load"chr$(34)"geos"chr$(34)",8,1[
    COM-7]":poke631,19:poke632,13
700 poke198,2:new
710 print"[SHIFT-CLR][SPACE]welkom[SPACE]!
720 print"met[SPACE]dit[SPACE]programm
    a[SPACE]kunt[SPACE]u[SPACE]uw[SPAC
    E]eigen[SPACE]geos-"
730 print"icons[SPACE]ontwerpen.het[SP
    ACE]programma[SPACE]bestaat"
740 print"uit[SPACE]2[SPACE]fasen:het[
    SPACE]grabben[SPACE]en[SPACE]het[S
    PACE]editen."
750 print"tijdens[SPACE]het[SPACE]grab
    ben[SPACE]kunt[SPACE]u[SPACE]een[S
    PACE]icon"
760 print"in[SPACE]het[SPACE]geheugen[
```

```
     SPACE]bewaren[SPACE]of[SPACE]selec
     teren."
770  print"tijdens[SPACE]het[SPACE]edit
     en[SPACE]kunt[SPACE]u[SPACE]de[SPA
     CE]icon[SPACE]met"
780  print"de[SPACE]cursor/return/space
     [SPACE]toetsen[SPACE]bewerken.";
790  print"de[SPACE]funkties[SPACE]1-7[
     SPACE]bij[SPACE]het[SPACE]editen[S
     PACE]wijzen"
800  print"zichzelf.als[SPACE]u[SPACE]e
     en[SPACE]icon[SPACE]wilt[SPACE]kop
     ieren"
810  print"naar[SPACE]een[SPACE]andere[
     SPACE]file[SPACE]dan[SPACE]plaats[
     SPACE]u[SPACE]dat"
820  print"icon[SPACE]tijdens[SPACE]het
     [SPACE]grabben[SPACE]in[SPACE]de[S
     PACE]buffer,"
830  print"en[SPACE]dan[SPACE]kopieert[
     SPACE]u[SPACE]het[SPACE]van[SPACE]
     de[SPACE]buffer"
840  print"terug[SPACE]op[SPACE]het[SPA
     CE]scherm[SPACE](6)[SPACE]en[SPACE
     ]dan[SPACE]schrijft"
850  print"u[SPACE]het[SPACE]weg[SPACE]
     (2)[SPACE]naar[SPACE]disk.[2xCRSR-
     DOWN]":goto70
```

** EINDE LISTING geosikon

Checksum geosikon

REGEL 10	81	REGEL 450	234
REGEL 20	3	REGEL 460	103
REGEL 30	51	REGEL 470	125
REGEL 40	55	REGEL 480	202
REGEL 50	217	REGEL 490	137
REGEL 60	33	REGEL 500	211
REGEL 70	245	REGEL 510	62
REGEL 80	54	REGEL 520	138
REGEL 90	173	REGEL 530	52
REGEL 100	61	REGEL 540	113
REGEL 101	149	REGEL 550	53
REGEL 110	113	REGEL 560	88
REGEL 120	38	REGEL 570	82
REGEL 130	210	REGEL 580	94
REGEL 140	133	REGEL 590	108
REGEL 150	176	REGEL 600	91
REGEL 160	238	REGEL 610	34
REGEL 170	120	REGEL 611	194
REGEL 180	201	REGEL 620	4
REGEL 190	212	REGEL 621	189
REGEL 200	144	REGEL 630	58
REGEL 210	52	REGEL 640	191
REGEL 220	97	REGEL 650	140
REGEL 230	144	REGEL 660	165
REGEL 240	80	REGEL 670	240
REGEL 250	2	REGEL 680	59
REGEL 260	235	REGEL 690	134
REGEL 270	34	REGEL 700	115
REGEL 280	221	REGEL 710	96
REGEL 290	88	REGEL 720	80
REGEL 291	135	REGEL 730	212
REGEL 300	209	REGEL 740	213
REGEL 310	113	REGEL 750	88
REGEL 320	94	REGEL 760	78
REGEL 330	93	REGEL 770	183
REGEL 340	205	REGEL 780	204
REGEL 350	127	REGEL 790	170
REGEL 360	114	REGEL 800	61
REGEL 370	55	REGEL 810	108
REGEL 380	254	REGEL 820	239
REGEL 390	72	REGEL 830	4
REGEL 400	186	REGEL 840	240
REGEL 410	161	REGEL 850	64
REGEL 420	23		
REGEL 430	67		
REGEL 440	141		

Datamak

Om met dit programma te kunnen werken heeft u een diskdrive nodig. Wanneer u een machinetaal-programma of sprites wilt omzetten in dataregels, moeten al deze data worden ingetikt. Afgezien van het vele werk kan dit ook veel ergernis geven, een fout is zo gemaakt. Dit programma kan dit probleem oplossen. De dataregels worden automatisch gegenereerd. Ook dit programma is van de hand van Wim de Wild uit Nigtevecht.

```
1    rem ******************************
***
2    rem **          data-maker
**
3    rem **          ----------
**
4    rem **     voor commodore-info
**
5    rem **       1989   pitsoft
**
6    rem ******************************
***
7    print"[SHIFT-CLR]"chr$(8)"[CTRL-N]
     ";:poke53281,0:poke53280,0:poke646
     ,8:goto15
8    print"[SPACE]Lezen[SPACE]met[SPACE
     ]:"
9    print"[SPACE][CTRL-9]1[CTRL-0][2xS
     PACE]Drive[SPACE]A[2xSPACE](8)"
10   print"[SPACE][CTRL-9]2[CTRL-0][2xS
     PACE]Drive[SPACE]B[2xSPACE](9)"
11   geta$:ifa$<>"1"anda$<>"2"then11
12   ifa$="1"thendl=8
13   ifa$="2"thendl=9
14   return
15   print"[COM-8]Wanneer[SPACE]u[SPACE
     ]uw[SPACE]machinetaalprogramma[SPA
     CE]of"
16   print"uw[SPACE]sprites[SPACE]wilt[
     SPACE]omzetten[SPACE]in[SPACE]data
     regels"
17   print"moet[SPACE]u[SPACE]al[SPACE]
     die[SPACE]data[SPACE]intikken.Het[
     SPACE]maken"
18   print"van[SPACE]fouten[SPACE]komt[
     SPACE]dan[SPACE]veel[SPACE]voor[SP
     ACE]en[SPACE]ze[SPACE]zijn";
19   print"er[SPACE]dan[SPACE]moeilijk[
     SPACE]uit[SPACE]te[SPACE]halen."
20   print"Dit[SPACE]programma[SPACE]lo
     st[SPACE]dat[SPACE]probleem[SPACE]
     voor[SPACE]u"
21   print"op.Het[SPACE]genereert[SPACE
     ]automatisch[SPACE]de[SPACE]data-"
22   print"regels.U[SPACE]kunt[SPACE]de
     [SPACE]data[SPACE]laten[SPACE]uitd
     raaien"
23   print"op[SPACE]de[SPACE]monitor[SP
     ACE]of[SPACE]printer[SPACE]maar[SP
     ACE]u[SPACE]kunt"
24   print"ze[SPACE]ook[SPACE]als[SPACE
     ]basic-programma[SPACE]laten[SPACE
     ]weg-"
25   print"schrijven[SPACE]op[SPACE]DIS
     K.U[SPACE]kunt[SPACE]het[SPACE]aan
```

```
     tal"
26   print"cijfers[SPACE]per[SPACE]rege
     l[SPACE]zelf[SPACE]bepalen[SPACE]e
     n[SPACE]het"
27   print"programma[SPACE]werkt[SPACE]
     met[SPACE]1[SPACE]of[SPACE]2[SPACE
     ]diskdrives."
28   print"[7xSPACE]<<<<[SPACE]druk[SPA
     CE]op[SPACE]een[SPACE]toets[SPACE]
     >>>>"
29   print"[CTRL-8][CRSR-DOWN]Gemaakt[S
     PACE]door:Wim[SPACE]de[SPACE]Wild"
30   wait198,1:poke198,0
31   input"[SHIFT-CLR]Geef[SPACE]de[SPA
     CE]FILENAAM[SPACE]";n$
32   input"hoeveel[SPACE]getallen[SPACE
     ]per[SPACE]regel[SPACE](1-16)[4xSP
     ACE]8[3xCRSR-LEFT]";ge
33   ifge<1orge>16then32
34   ge=ge-1
35   print"[2xCRSR-DOWN][COM-3]OUTPUT[S
     PACE]naar:"
36   print"------------"
37   print"[CRSR-DOWN][SPACE][CTRL-9]b[
     CTRL-0]eeldscherm"
38   print"[CRSR-DOWN][SPACE][CTRL-9]p[
     CTRL-0]rinter"
39   print"[CRSR-DOWN][SPACE][CTRL-9]d[
     CTRL-0]rive"
40   geta$:ifa$=""then40
41   ifa$="b"thende=3:goto45
42   ifa$="p"thende=4:goto45
43   ifa$="d"then52
44   goto40
45   gosub8:open4,de:poke198,0:gosub70
46   open1,dl,0,n$:poke144,0
47   get#1,a$,b$:print#4,"beginadres[SP
     ACE]:"asc(b$)*256+asc(a$+chr$(0))
48   x=1000
49   b$=str$(x)+"[SPACE]data"
50   gosub75:ifkk<>1thenx=x+1:print#4,b
     $:ifpeek(198)=0then49
51   close4:close1:wait198,1:poke198,0:
     run
52   input"[SHIFT-CLR][SPACE]Filenaam[S
     PACE]om[SPACE]te[SPACE]schrijven";
     n2$
53   gosub8:print"[SPACE]Schrijven[SPAC
     E]naar:"
54   print"[SPACE][CTRL-9]1[CTRL-0][2xS
     PACE]Drive[SPACE]A[2xSPACE](8)"
55   print"[SPACE][CTRL-9]2[CTRL-0][2xS
     PACE]Drive[SPACE]B[2xSPACE](9)"
56   geta$:ifa$<>"1"anda$<>"2"then56
57   ifa$="1"thendv=8
58   ifa$="2"thendv=9
59   input"Startadres[4xSPACE]2049[6xCR
     SR-LEFT]";b
60   ifb<2049orb>40960then59
61   gosub70:open1,dl,0,n$:poke144,0:ge
     t#1,a$,b$
62   print"Beginadres:";asc(b$+chr$(0))
     *256+asc(a$+chr$(0))
63   open2,dv,2,"`:"+n2$+",p,w":print#2
     ,chr$(1);:print#2,chr$(8);
64   b$="":input"regelnummer[SPACE]0-30
     000";x:ifx<0orx>30000then64
65   gosub76:ifkk=1thenprint#2,chr$(0);
     chr$(0);:close2:close1:run
66   b=b+len(b$)+3:p=b:gosub79:print#2,
     chr$(l);:print#2,chr$(h);:p=x:gosu
     b79
67   print#2,chr$(l);:print#2,chr$(h);:
     print#2,chr$(131);
68   forw=1tolen(b$)
69   print#2,mid$(b$,w,1);:next:print#2
     ,chr$(0);:x=x+1:b$="":b=b+2:goto65
70   rem ** disk openen **
71   poke144,0:open1,dl,0,n$:close1
72   open1,dl,15:input#1,a,b$,c,d:close
     1
73   ifb$<>"ok"thenprinta,b$,c,d:end
74   return
75   rem ** data-string halen van disk
     **
76   kk=0:fork=0toge:get#1,a$:a$=str$(a
     sc(a$+chr$(0))):a$=right$(a$,len(a
     $)-1)
77   b$=b$+a$+",":ifst=66thenkk=1:retur
     n
78   next:b$=left$(b$,len(b$)-1):return
79   rem **     hi-lo     **
80   h=int(p/256):l=p-256*h:return
```

** EINDE LISTING datamak

Checksum Datamak

REGEL 1	249	REGEL 44	237
REGEL 2	238	REGEL 45	132
REGEL 3	249	REGEL 46	76
REGEL 4	123	REGEL 47	122
REGEL 5	59	REGEL 48	203
REGEL 6	249	REGEL 49	141
REGEL 7	170	REGEL 50	213
REGEL 8	251	REGEL 51	61
REGEL 9	246	REGEL 52	217
REGEL 10	249	REGEL 53	100
REGEL 11	0	REGEL 54	246
REGEL 12	56	REGEL 55	249
REGEL 13	58	REGEL 56	9
REGEL 14	142	REGEL 57	66
REGEL 15	91	REGEL 58	68
REGEL 16	220	REGEL 59	64
REGEL 17	56	REGEL 60	10
REGEL 18	187	REGEL 61	204
REGEL 19	188	REGEL 62	118
REGEL 20	240	REGEL 63	137
REGEL 21	190	REGEL 64	66
REGEL 22	225	REGEL 65	182
REGEL 23	158	REGEL 66	81
REGEL 24	254	REGEL 67	120
REGEL 25	42	REGEL 68	217
REGEL 26	213	REGEL 69	136
REGEL 27	20	REGEL 70	39
REGEL 28	1	REGEL 71	87
REGEL 29	26	REGEL 72	246
REGEL 30	96	REGEL 73	223
REGEL 31	102	REGEL 74	142
REGEL 32	95	REGEL 75	205
REGEL 33	91	REGEL 76	117
REGEL 34	166	REGEL 77	25
REGEL 35	226	REGEL 78	157
REGEL 36	249	REGEL 79	144
REGEL 37	176	REGEL 80	38
REGEL 38	182		
REGEL 39	12		
REGEL 40	49		
REGEL 41	105		
REGEL 42	120		
REGEL 43	56		

rapport

```
100  print"[SHIFT-CLR]":a=1:dimb(40),s(
     40):b1=1388:poke53280,0:poke53281,
     0:poke646,15
110  x=4:fory=0to9:gosub360:print"[SHIF
     T -]":next
120  print"[4xCRSR-RIGHT][COM-Z][34xSHI
     FT *]"
130  x=0:fory=0to9:gosub360:print10-y:n
     ext
140  x=0:y=12:gosub360:print"typ[SPACE]
     een[SPACE]'[PIJL LINKS]'[SPACE]ach
     ter[SPACE]het[SPACE]laatste[SPACE]
     cijfer"
150  x=0:y=14:gosub360:print"cijfer"a:i
     nput"[CRSR-UP][9xCRSR-RIGHT]";a$
160  ifa$="[PIJL LINKS]"anda=1then150
170  ifa$="[PIJL LINKS]"thena=a-1:goto2
     10
180  ifval(a$)<1orval(a$)>10thenprint"[
     CRSR-UP][10xCRSR-RIGHT][14xSPACE]"
     :goto150
190  ifright$(a$,1)="[PIJL LINKS]"thenb
     (a)=val(left$(a$,len(a$)-1)):goto2
     10
200  b(a)=val(a$):a=a+1:ifa<35then150
210  forw=1toa:t=b1+w-(int(b(w)+.5)-1)*
     40:forg=b1+wtotstep-40:pokeg,160
220  pokeg+54272,int(b(w)+.5):nextg,w
230  ifa=35thena=34
240  h=0:l=10:r=0:forf=1toa:ifb(f)>hthe
     nh=b(f)
250  ifb(f)<lthenl=b(f)
260  r=r+b(f):next:g=r/a
270  gs=0:forf=1toa:s(f)=(b(f)-g)[KWAD
     RAATPIJL] 2:gs=gs+s(f)/a:next:sd=
     sqr(gs)
280  print"[COM-3]uitslag:[CTRL-8]"
290  print"aantal[SPACE]cijfers:"a
300  print"laagste[SPACE]cijfer:"l
310  print"hoogste[SPACE]cijfer:"h
320  print"standaard[SPACE]dev.:"sd
330  print"gemiddelde[4xSPACE]:"g
340  print"afgerond[6xSPACE]:"int(g+.5)
350  poke198,0:wait198,1:end
360  poke211,x:poke214,y:sys58732:retur
     n
```

** EINDE LISTING rapport

Checksum Rapport

REGEL 100	78	REGEL 240	11
REGEL 110	33	REGEL 250	225
REGEL 120	126	REGEL 260	8
REGEL 130	97	REGEL 270	157
REGEL 140	52	REGEL 280	100
REGEL 150	136	REGEL 290	15
REGEL 160	85	REGEL 300	23
REGEL 170	82	REGEL 310	43
REGEL 180	183	REGEL 320	77
REGEL 190	84	REGEL 330	34
REGEL 200	168	REGEL 340	183
REGEL 210	236	REGEL 350	26
REGEL 220	59	REGEL 360	69
REGEL 230	231		

Super disk

Het programma Superdisk is een utility voor de diskdrive bezitters. Het is een handig programma om eens op een regenachtige dag je diskettes bij te werken. De meest voorkomende commando's zijn in het programma opgenomen: onder andere Format, Scratch, Valide, Rename, enzovoort. Het programma is menugestuurd en tekst en uitleg zijn in het programma opgenomen. De maker is Marco Vigelius uit Dordrecht.

```
10   rem ******************************
     **
12   rem  *  super-disk  commodore 64
     *
14   rem   *      gemaakt voor        *
16   rem    *    commodore infc=      *
18   rem     *        door:           *
20   rem      *    marco vigelius    *
22   rem       *                     *
24   rem        *   (c) 1989/90    *
26   rem         ****************
28   poke53281,12:poke53280,11:print"[S
     HIFT CLR]"
30   print"[CTRL-9][CTRL-2][CRSR-UP][12
     xSPACE]super[SPACE]disk-menu[13xSP
     ACE][CTRL-0]"
32   gosub314
34   print"[CRSR-UP]"
36   print"[10xSPACE][CTRL-8](1)[2xSPAC
     E][CTRL-3]directory":print
38   print"[10xSPACE][CTRL-8](2)[2xSPAC
     E][CTRL-2]format[3xSPACE]":print
40   print"[10xSPACE][CTRL-8](3)[2xSPAC
     E][CTRL-7]rename[SPACE]disk":print
42   print"[10xSPACE][CTRL-8](4)[2xSPAC
     E][CTRL-3]scratch[2xSPACE]":print
44   print"[10xSPACE][CTRL-8](5)[2xSPAC
     E][CTRL-2]rename[SPACE]program":pr
     int
46   print"[10xSPACE][CTRL-8](6)[2xSPAC
     E][CTRL-7]validate":print
48   print"[10xSPACE][CTRL-8](7)[2xSPAC
     E][CTRL-5]stoppen":print
50   print"[10xSPACE][CTRL-8](8)[2xSPAC
     E][CTRL-4]help":print:print:gosub3
     12:print
52   print"[CTRL-8][5xSPACE]maak[SPACE]
     uw[SPACE]keuze[SPACE]?"
54   geta:ifa<1ora>8then54
56   onagoto58,80,108,158,180,198,216,2
     28
58   print"[SHIFT-CLR][CTRL-9][CTRL-2][
     10xSPACE]super-disk[SPACE]director
     y[10xSPACE]"
60   gosub314
62   print"[SPACE]doe[SPACE]de[SPACE]di
     sk[SPACE]in[SPACE]de[SPACE]disk-dr
     ive"
64   print:print"[SPACE]druk[SPACE]op[S
     PACE]een[SPACE]toets[8xSPACE]"
66   geta$:ifa$=""then66
68   print"[5xCRSR-UP]"
70   fort=1to6:print"[39xSPACE]":nextt
72   gosub320
74   print:print"[CTRL-8]druk[SPACE]een
```

```
    [SPACE]toets[SPACE]!"
76  geta$:ifa$=""then76
78  run
80  print"[SHIFT-CLR][CTRL-9][CTRL-2][
    11xSPACE]super-disk[SPACE]format[1
    2xSPACE]"
82  gosub314
84  print"[SPACE]wilt[SPACE]u[SPACE]ee
    n[SPACE]disk[SPACE]formatteren[SPA
    CE]?[SPACE](j/n)"
86  geta$:ifa$<>"j"anda$<>"n"then86
88  ifa$="n"thenrun
90  print:print"doe[SPACE]de[SPACE]dis
    k[SPACE]die[SPACE]geformatteerd[SP
    ACE]moet"
92  print"in[SPACE]de[SPACE]disk-drive
    !"
94  print:input"diskette-naam";a$
96  print:input"diskette-[SPACE]id[SPA
    CE]";b$:b$=str$(val(b$))
98  nextp:a$="new:"+a$+","+b$:open1,8,
    15,a$:close1
100 print:print"wilt[SPACE]u[SPACE]nog
    [SPACE]een[SPACE]disk[SPACE]format
    teren[SPACE]?[SPACE]"
102 getc$:ifc$<>"j"andc$<>"n"then102
104 ifc$="j"then80
106 run
108 print"[SHIFT-CLR][CTRL-9][CTRL-2][
    9xSPACE]super-disk[SPACE]rename[SP
    ACE]disk[9xSPACE]"
110 gosub314
112 forl=1to13:print"";:next:print:pri
    nt:x$="0"
114 dv=val(x$)
116 dv$=x$:ifdvand254thenprint"[6xCRSR
    -UP]":goto112
118 print"verander[SPACE]de[SPACE]disk
    [SPACE][CTRL-9]n[CTRL-0]aam":print
    :printtab(9)"of[SPACE]de[SPACE]dis
    k[SPACE][CTRL-9]i[CTRL-0]d[SPACE]?
    ";
120 gosub148:ifx$<>"n"andx$<>"i"then12
    0
122 printx$:open15,8,15:print#15,"i"+d
    v$:open1,8,3,"#":print#15,"u1:3"dv
    ",18,0"
124 ifx$="n"then138
126 print#15,"b-p:3,162":get#1,a$,b$,c
    $,d$,e$:print"nu[SPACE]is[SPACE]de
    [SPACE]disk[SPACE]id[SPACE][CTRL-9
    ]"a$;
128 printb$;c$;d$;e$:input"wat[SPACE]i
    s[SPACE]het[SPACE]nieuwe[SPACE]id?
    [SPACE](max.5[SPACE]karakters)";ni
    $
130 iflen(ni$)<>4theni$=left$(ni$+"[SP
    ACE]",5)
132 print#15,"b-p:3,162":print#1,ni$;:
    print#15,"u2:3"dv",18,0
134 i$="i"+mid$(str$(dv),2):print#15,i
    $
136 goto150
138 print#15,"b-p:3,144":dn$="":forl=1
    to16:get#1,a$:dn$=dn$+a$:next
140 print"nu[SPACE]is[SPACE]de[SPACE]d
    isknaam:":print:print"[CTRL-9][SPA
    CE]"+dn$+"[SPACE]"
142 print"wat[SPACE]is[SPACE]de[SPACE]
    nieuwe[SPACE]disknaam[SPACE](max.1
    6[SPACE]karak-ters!)":inputnd$:pri
    nt
144 nd$=left$(nd$+"[14xSPACE]",16)
146 print#15,"b-p:3,144":print#1,nd$;:
    print#15,"u2:3"dv",18,0":goto134
148 getx$:return
150 print:print"[SPACE][CTRL-8]wilt[SP
    ACE]u[SPACE]nog[SPACE]iets[SPACE]v
    eranderen[SPACE]?"
152 getd$:ifd$<>"j"andd$<>"n"then152
154 ifd$="j"thenclose1:close15:goto108
156 run
158 print"[SHIFT-CLR][CTRL-9][CTRL-2][
    12xSPACE]super-disk[SPACE]scratch[
    10xSPACE]"
160 gosub314
162 print:print
164 print"weet[SPACE]u[SPACE]welk[SPAC
    E]programma[SPACE]u[SPACE]wilt[SPA
    CE]scratchen[SPACE]?":print
166 geta$:ifa$<>"j"anda$<>"n"then166
168 ifa$="n"thenrun
170 input"welk[SPACE]programma";a$:a$=
    "scratch:"+a$:open1,8,15,a$:close1
172 print:print"wilt[SPACE]u[SPACE]nog
    [SPACE]een[SPACE]programma[SPACE]s
    cratchen[SPACE]?"
174 geta$:ifa$<>"j"anda$<>"n"then174
176 ifa$="j"then158
178 run
180 print"[SHIFT-CLR][CTRL-9][CTRL-2][
    7xSPACE]super-disk[SPACE]rename[SP
    ACE]program[8xSPACE]"
182 gosub314
184 print:print
186 input"oude[SPACE]naam";a$
188 print:input"nieuwe[SPACE]naam";b$:
    a$="r:"+b$+"="+a$:open1,8,15,a$:cl
    ose1
190 print:print"wilt[SPACE]u[SPACE]nog
    [SPACE]een[SPACE]programma[SPACE]v
    an[SPACE]naam[SPACE]veran-deren[SP
    ACE]?"
192 geta$:ifa$<>"j"anda$<>"n"then192
194 ifa$="j"then180
196 run
198 print"[SHIFT-CLR][CTRL-9][CTRL-2][
    11xSPACE]super-disk[SPACE]validate
    [10xSPACE]"
200 gosub314
202 print:print"doe[SPACE]de[SPACE]dis
    k[SPACE]in[SPACE]de[SPACE]drive[SP
    ACE]en[SPACE]!!"
204 print:print"druk[SPACE]'j'[SPACE]v
    oor[SPACE]'validate'
206 print:print"druk[SPACE]'n'[SPACE]v
    oor[SPACE]menu"
208 geta$:ifa$<>"j"anda$<>"n"then208
210 ifa$="j"then214
212 run
214 open1,8,15,"validate:":close1:run
216 print"[SHIFT-CLR][CTRL-9][CTRL-2][
    11xSPACE]super-disk[SPACE]stoppen[
    11xSPACE]"
218 gosub314
220 print:print:print"[11xSPACE][CTRL-
```

```
     8]zeker[SPACE]weten[SPACE]?[SPACE]
     (j/n)[SPACE]"
222  geta$:ifa$<>"j"anda$<>"n"then222
224  ifa$="n"thenrun
226  sys64738
228  print"[SHIFT-CLR][CTRL-9][CTRL-2][
     12xSPACE]super-disk[SPACE]help[13x
     SPACE]"
230  gosub314
232  print"[CTRL-0][CTRL-3][SPACE]direc
     tory[4xSPACE][CTRL-8][SPACE]:laat[
     SPACE]u[SPACE]de[SPACE]inhoud[SPAC
     E]van"
234  print"[CTRL-8][16xSPACE]een[SPACE]
     diskette[SPACE]zien."
236  print:print"[CTRL-2][SPACE]format[
     7xSPACE][CTRL-8][SPACE]:deze[SPACE
     ]functie[SPACE]laat[SPACE]u"
238  print"[CTRL-8][16xSPACE]een[SPACE]
     diskette[SPACE]formatteren"
240  print"[CTRL-8][CRSR-UP][16xSPACE]p
     as[SPACE]op!als[SPACE]er[SPACE]al[
     SPACE]pro-"
242  print"[CTRL-8][16xSPACE]gramma's[S
     PACE]opstaan,gaan"
244  print"[CTRL-8][16xSPACE]deze[SPACE
     ]'allemaal'[SPACE]verloren"
246  print"[CTRL-7][SPACE]rename[SPACE]
     disk[2xSPACE][CTRL-8][SPACE]:hierm
     ee[SPACE]kunt[SPACE]u[SPACE]een"
248  print"[CTRL-8][16xSPACE]diskette[S
     PACE]van[SPACE]naam[SPACE]ver-"
250  print"[CTRL-8][16xSPACE]anderen,en
     [SPACE]ook[SPACE]het[SPACE]'id'"
252  print:print"[CTRL-3][SPACE]scratch
     [6xSPACE][CTRL-8][SPACE]:laat[SPAC
     E]u[SPACE]een[SPACE]programma"
254  print"[CTRL-8][16xSPACE]van[SPACE]
     de[SPACE]diskette[SPACE]verwij-"
256  print"[CTRL-8][16xSPACE]deren.als[
     SPACE]u[SPACE]'bv.'d*'[2xSPACE]"
258  print"[CTRL-8][16xSPACE]typt,worde
     n[SPACE]alle[SPACE]files"
260  print"[CTRL-8][16xSPACE]die[SPACE]
     met[SPACE]een[SPACE]'d'[SPACE]begi
     nnen"
262  print"[CTRL-8][CRSR-UP][16xSPACE]v
     erwijderd.
264  print:print"[8xSPACE][CTRL-9][CTRL
     7]<<<<[SPACE]druk[SPACE]een[SPACE
     ]toets[SPACE]>>>>[CTRL-0][3xCRSR-UP]"
266  geta$:ifa$=""then266
268  print"[SHIFT-CLR][CTRL-9][CTRL-2][
     12xSPACE]super-disk[SPACE]help[13x
     SPACE]"
270  gosub314
272  print"[CTRL-0][CTRL-3][SPACE]renam
     e[SPACE]program[CTRL-8][SPACE]:hie
     rmee[SPACE]kunt[SPACE]u"
274  print"[CTRL-8][17xSPACE]een[SPACE]
     programma[SPACE]van[SPACE]naam"
276  print"[CTRL-8][17xSPACE]veranderen
     .door[SPACE]alleen"
278  print"[CTRL-8][17xSPACE]op[SPACE]<
     return>[SPACE]te[SPACE]drukken"
280  print"[CTRL-8][17xSPACE]gebeurt[SP
     ACE]er[SPACE]niets."
282  print:print"[CTRL-2][SPACE]validat
     e[6xSPACE][CTRL-8][SPACE]:dit[SPAC
     E]commando[SPACE]"
284  print"[CTRL-8][17xSPACE]controleer
     d[SPACE]uw[SPACE]diskette"
286  print"[CTRL-8][CRSR-UP][17xSPACE]e
     n[SPACE]verwijdert[SPACE]al[SPACE]
     de[SPACE]"
288  print"[CTRL-8][17xSPACE]mislukte[S
     PACE]files.(die[SPACE]met"
290  print"[CTRL-8][CRSR-UP][17xSPACE]0
     [SPACE]blocks)
292  print:print"[CTRL-7][SPACE]stoppen
     [7xSPACE][CTRL-8][SPACE]:hiermee[S
     PACE]kunt[SPACE]u[SPACE]dit"
294  print"[CTRL-8][17xSPACE]programma[
     SPACE]beeindigen.u"
296  print"[CTRL-8][17xSPACE]kunt[SPACE
     ]het[SPACE]weer[SPACE]starten"
298  print"[CTRL-8][18xSPACE]door[SPACE
     ]'run'[SPACE]gevolgd[SPACE]met"
300  print"[CTRL-8][CRSR-UP][17xSPACE]<
     return>[SPACE]te[SPACE]geven."
302  print:print"[CTRL-3][SPACE]help[10
     xSPACE][CTRL-8][SPACE]:geeft[SPACE
     ]uitleg[SPACE]over"
304  print"[CTRL-8][17xSPACE]alle[SPACE
     ]commando's."
306  print"[8xSPACE][CTRL-9][CTRL-7]<<<
     <[SPACE]druk[SPACE]een[SPACE]toets
     [SPACE]>>>>[4xCRSR-UP]"
308  geta$:ifa$=""then308
310  run
312  open1,8,15:input#1,a$,b$,c$,d$:pri
     nt"[CTRL-1][5xSPACE]situatie[SPACE
     ]disk:[SPACE]"b$:close1:return
314  print"[CTRL-9][CTRL-2][CRSR-UP][7x
     SPACE]by[SPACE]marco[SPACE]vigeliu
     s[SPACE](c)1990[8xSPACE]"
316  print"[CTRL-0][CTRL-2][CRSR-UP][40
     xCOM-T][CTRL-8]"
318  return
320  rem ******directory********
322  open15,8,15:print#15,"i0"
324  print"[HOME][3xCRSR-DOWN]";:open1,
     8,0,"$0"
326  get#1,a$,b$
328  get#1,a$,b$
330  get#1,a$,b$
332  c=0:ifa$<>""thenc=asc(a$)
334  ifb$<>""thenc=c+asc(b$)*256
336  print"[CTRL-1]"mid$(str$(c),2);tab
     (5);"[CTRL-8]";
338  get#1,b$:ifst<>0thengoto 356
340  ifb$<>chr$(34)then338
342  get#1,b$:ifb$<>chr$(34)thenprintb$
     ;:l=l+1:goto342
344  get#1,b$:ifb$=chr$(32)then344
346  c$=""
348  c$=c$+b$:get#1,b$:ifb$<>""then348
350  forkk=0to17-l:print"[SPACE]";:next
     :print"[CTRL-7]"left$(c$,3)
352  get k$:ifk$<>""then:poke198,0:wait
     198,1:poke198,0
354  ifst=0thenl=0:goto328
356  print"[COM-2]blocks[SPACE]free":cl
     ose1:close15
358  return
```

** EINDE LISTING superd

Checksum superd

REGEL 100	78
REGEL 110	33
REGEL 120	126
REGEL 130	97
REGEL 140	52
REGEL 150	136
REGEL 160	85
REGEL 170	82
REGEL 180	183
REGEL 190	84
REGEL 200	168
REGEL 210	236
REGEL 220	59
REGEL 230	2
REGEL 10	207
REGEL 12	217
REGEL 14	35
REGEL 16	229
REGEL 18	81
REGEL 20	189
REGEL 22	227
REGEL 24	234
REGEL 26	47
REGEL 28	52
REGEL 30	51
REGEL 32	37
REGEL 34	110
REGEL 36	161
REGEL 38	159
REGEL 40	212
REGEL 42	247
REGEL 44	169
REGEL 46	62
REGEL 48	155
REGEL 50	162
REGEL 52	4
REGEL 54	182
REGEL 56	11
REGEL 58	35
REGEL 60	37
REGEL 62	91
REGEL 64	236
REGEL 66	57
REGEL 68	178
REGEL 70	185
REGEL 72	34
REGEL 74	12
REGEL 76	58
REGEL 78	138
REGEL 80	55
REGEL 82	37
REGEL 84	19
REGEL 86	65
REGEL 88	101
REGEL 90	12
REGEL 92	240
REGEL 94	227
REGEL 96	55
REGEL 98	96
REGEL 100	178
REGEL 102	108
REGEL 104	65
REGEL 106	138
REGEL 108	81
REGEL 110	37
REGEL 112	72
REGEL 114	222
REGEL 116	214
REGEL 118	140
REGEL 120	183
REGEL 122	109
REGEL 124	142
REGEL 126	182
REGEL 128	37
REGEL 130	219
REGEL 132	205
REGEL 134	79
REGEL 136	31
REGEL 138	29
REGEL 140	113
REGEL 142	13
REGEL 144	184
REGEL 146	69
REGEL 148	229
REGEL 150	37
REGEL 152	116
REGEL 154	71
REGEL 156	138
REGEL 158	118
REGEL 160	37
REGEL 162	108
REGEL 164	130
REGEL 166	112
REGEL 168	101
REGEL 170	196
REGEL 172	129
REGEL 174	111
REGEL 176	117
REGEL 178	138
REGEL 180	62
REGEL 182	37
REGEL 184	108
REGEL 186	179
REGEL 188	46
REGEL 190	255
REGEL 192	111
REGEL 194	112
REGEL 196	138
REGEL 198	184
REGEL 200	37
REGEL 202	171
REGEL 204	58
REGEL 206	253
REGEL 208	109
REGEL 210	110
REGEL 212	138
REGEL 214	137
REGEL 216	151
REGEL 218	37
REGEL 220	124
REGEL 222	105
REGEL 224	101
REGEL 226	170
REGEL 228	151
REGEL 230	37
REGEL 232	196
REGEL 234	20
REGEL 236	3
REGEL 238	247
REGEL 240	210
REGEL 242	3
REGEL 244	151
REGEL 246	37
REGEL 248	244
REGEL 250	220
REGEL 252	161
REGEL 254	74
REGEL 256	245
REGEL 258	88
REGEL 260	227
REGEL 262	14
REGEL 264	171
REGEL 266	107
REGEL 268	151
REGEL 270	37
REGEL 272	201
REGEL 274	251
REGEL 276	120
REGEL 278	33
REGEL 280	209
REGEL 282	6
REGEL 284	197
REGEL 286	187
REGEL 288	106
REGEL 290	1
REGEL 292	71
REGEL 294	110
REGEL 296	242
REGEL 298	224
REGEL 300	162
REGEL 302	62
REGEL 304	143
REGEL 306	215
REGEL 308	104
REGEL 310	138
REGEL 312	172
REGEL 314	97
REGEL 316	27
REGEL 318	142
REGEL 320	144
REGEL 322	28
REGEL 324	236
REGEL 326	24
REGEL 328	24
REGEL 330	24
REGEL 332	15
REGEL 334	232
REGEL 336	210
REGEL 338	85
REGEL 340	25
REGEL 342	49
REGEL 344	35
REGEL 346	93
REGEL 348	106
REGEL 350	31
REGEL 352	252
REGEL 354	73
REGEL 356	157
REGEL 358	142

Disk Driver

Disk driver is een krachtige disk-utility. Het is geschreven voor die mensen, die al ver gevorderd zijn in het gebruik van de disk drive. Disk driver kan namelijk een willekeurige sector ingeladen worden, en daarmee kunnen legio bewerkingen mee worden uitgevoerd. Van alle functies volgt nu een korte uitleg.

CRSR-toetsen - besturing van de cursor
HOME- cursor op positie 0
Shift Z - instel-track +1
Z - instel-track -1
Shift X - instel-sector +1
X - instel-sector -1
CLR - instel-track 18, instel-sector 0 (BAM)
F5 - sector lezen
- sector +1
- sector -1
L - lees link (staat in byte 0-1 van de sector)
K - lees vorige track/sector)
S - schrijf sector over oude heen. Met de instel-toetsen kan desgewenst de sector naar een andere plaats geschreven worden.
T - tekst invoeren
F7 regel - regel tekst invoeren (maximaal 16 tekens die (meerdere malen) in de sector op het scherm plaatsen
F4 - vorige regel terugroepen
A - verander waarde van karakter onder de cursor
V - vul sector op het scherm met ingevoerde waarde
***** - zoekfunctie: deze functie laat de tekens die een ASCII-waarde hebben van tussen de twee ingevoerde getallen wit oplichten, de rest wordt grijs. Het eerste getal

moet kleiner zijn dan het
twee. Een enkele waarde
laten oplichten: twee
opeenvolgende getallen invoeren
H - roep laatst ingelezen sector terug
E - disk drive RESET en terug naar BASIC

Bij het schrijven zal de bekende formaat-truc worden genegeerd. Slechts als er een stickertje op de disk zit, zal men niet op de disk kunnen schrijven.

Enkele toepassingsmogelijkheden zijn:
- Het repareren en veranderen van files
- Het terugroepen van files, mits ze niet overschreven zijn (1e byte van elke directory-naam veranderen in waarde 130, voor een PRG File).
- Het maken en veranderen van directories
- Het bekijken van de Block Availability Map (BAM), directory en file om ze te leren begrijpen
- Het markeren van diskettes (uw naam erop zetten)
- Het beveiligen van diskettes (byte 2 in track 18, sector 0, de formaataanduiding veranderen. Met Disk Driver kan dan nog wel op de disk worden geschreven.). De makers van dit programma zijn: Joris van den Heuvel en Hanno Krijgsman uit Terheijden.

```
10   rem disk driver / c=64
20   rem
30   rem joris van den heuvel
40   rem
50   :
100  rem 01 initialisatie
110  poke53265,11:print"[CTRL-2][SHIFT-
     CLR]"chr$(14)chr$(8);:b=53280:s=10
     24:it=18
120  x=53254:y=53255:cc=53269:e=49207:f
     =53157
130  o$="[CRSR-LEFT][CTRL-0][SPACE][CRS
     R-LEFT]":p$="[CRSR-LEFT][SPACE]":q
     $="[CRSR-LEFT][2xSPACE]":r$="1":s$
     =chr$(1)
140  pokeb,15:pokeb+1,0:pokex,23:pokey,
     49:poke808,234:poke650,128
150  deffna(l)=l*((peek(y)-49)/8)+((pee
     k(x)-23)/8)
160  gosub2050:pokef,13:sys52992:poke24
     9,1:poke250,0:poke49261,1
170  pokex+2,134:pokey+2,185:poke53277,
     16
180  pokex-6,103:pokey-6,201:pokex-4,11
     9:pokey-4,185
190  pokex-2,103:pokey-2,185
200  fort=0to4:poke53287+t,15:poke2040+
     t,13:next
210  gosub2230:gosub760:pokeb,0:pokecc,
     23:poke53265,27:goto1990
220  rem02 toets/hoofdlus
230  printq$(0)"[3xCRSR-RIGHT]"fna(16)q
     $
240  q=fna(40):poke1794,peek(s+q):print
     "[CRSR-DOWN][3xCRSR-RIGHT]"peek(s+
     q)q$
250  gett$:ift$=""ort$="[CRSR-RIGHT]"or
     t$="[CRSR-LEFT]"ort$="[CRSR-UP]"or
     t$="[CRSR-DOWN]"then230
260  a=asc(t$)
270  ifa=218ora=90ora=216ora=88thengosu
     b680:goto250
280  ift$="[HOME]"thenp=0:gosub530:goto
     230
290  op=p:p=fna(16)
300  ift$="."ort$=","ort$="l"ort$="k"or
     a=135thenot=mt:os=ms
310  ifa=135thenmt=it:ms=is:gosub810:po
     kecc,31:goto230
320  ift$="."andms<n(mt)thenms=ms+1:gos
     ub810:goto230
330  ift$=","andms>0thenms=ms-1:gosub81
     0:goto230
340  ift$="l"andls<=n(mt)andlt<36andlt>
     0thenmt=lt:ms=ls:gosub810:goto250
350  ift$="k"andi=2thenmt=vt:ms=vs:gosu
     b810:goto250
360  ift$="t"ort$="a"thenpoke250,0:n=1:
     gosub1940
370  ift$="t"thengosub1310:n=1:gosub194
     0:poke250,si:gosub1970:goto230
380  ift$="[SHIFT-CLR]"thenit=18:is=0:g
     osub760:goto250
390  ift$="a"thengosub1270:n=1:gosub194
     0:poke250,si:gosub1970:goto230
400  ift$="v"thengosub1660:goto230
410  ift$="s"thengosub570:n=1:gosub1940
     :poke250,si:gosub1970:goto250
420  ift$="h"thensyse:goto230
430  ift$="*"thengosub1720:goto230
440  ift$="e"thengosub470:pokecc,23:got
     o230
450  poke1714,63:fort=1to300:next:poke1
     714,32:poke198,0:goto230
460  rem03 eindigen
470  printq$(2)"[3xCRSR-DOWN]Eindigen[S
     PACE]?";:pokef,80
480  gett$:ift$="j"ort$="n"then500
490  goto480
500  ift$="n"thenprint"[CRSR-UP]":print
     "[10xSPACE]";:pokef,13:return
510  poke53265,11:print"[SHIFT-CLR]":op
     en15,8,15:print#15,"uj":close15:sy
     s65126
520  rem04 cursor positioneren
530  ifp>255thenp=op
540  cy=int(p/16):cx=23+8*(p-16*cy)
550  pokex,cx:pokey,49+8*cy:return
560  rem05 sector schrijven
570  printq$(2)"[CTRL-2]Schrijf[SPACE]s
     ector":print"Maak[SPACE]evt[SPACE]
     Instel"
580  pokecc,23:poke250,0
590  gosub1170:gosub680
600  ift$="j"then640
610  a=asc(t$):ifa=218ora=90ora=216ora=
     88then590
620  ifa=147thenit=18:is=0:gosub760:got
     o590
630  pokecc,31:goto660
640  ot=mt:os=ms:mt=it:ms=is
650  r$="2":gosub810:r$="1"
660  gosub1210:poke250,si:return
670  rem06 instel-t+s lus
680  ift$="Z"thenit=it+1:ifis>n(it)then
     is=n(it)
```

```
690 ift$="z"thenit=it-1
700 ift$="X"thenis=is+1
710 ift$="x"thenis=is-1
720 ifit>35thenit=1
730 ifit<1thenit=35:ifis>n(it)thenis=n
    (it)
740 ifis>n(it)thenis=0
750 ifis<0thenis=n(it)
760 printq$(2);:ifit>9theni$="":goto78
    0
770 i$="[SPACE]"
780 printspc(30)i$it;p$is;p$
790 return
800 rem07 lees/schrijfroutine
810 gosub1240:pokecc,0:printq$(1)tab(1
    2);:ifr$="1"thenprint"LOAD":goto83
    0
820 print"SAVE"
830 open15,8,15:open5,8,5,"#"
840 ifr$="2"thenn=2:gosub1930:sys49185
850 print#15,"u";r$;5;0;mt;ms
860 input#15,en,e$,et,es
870 ifen<20then940
880 poke198,0:print#15,"ui":close5:clo
    se15:gosub1210:printq$(1)tab(12)"[
    CTRL-2]FOUT"
890 print"[CRSR-DOWN]DOS[SPACE]melding
    "en":[COM-8]":print"[CTRL-9]"left$
    (e$,16)
900 print"[CTRL-9]"mid$(e$,17,len(e$))
    :ifet>0ores>0thenprint"[CTRL-2]Tr"
    et"[2xSPACE]Sr"es;
910 gett$:ift$=""then910
920 printq$(1)tab(12)"[4xSPACE]";:gosu
    b1210:ifi=0thenit=18:is=0:gosub760
    :goto1990
930 mt=ot:ms=os:goto960
940 ifr$="1"thensys49163:n=0:gosub1970
    :vt=ot:vs=os
950 it=mt:is=ms:close5:close15:printq$
    (1)tab(12)"-OK-"
960 ifi=1theni=2
970 ifen<21thenlt=peek(s):ls=peek(s+1)
980 iflt>199thenlt=lt-100
990 iflt>99thenlt=lt-100
1000 ifls>199thenls=ls-100
1010 ifls>99thenls=ls-100
1020 ifmt>9thenm$="":goto1040
1030 m$="[SPACE]"
1040 iflt>9thenl$="":goto1060
1050 l$="[SPACE]"
1060 ifvt>9thenv$="":goto1080
1070 v$="[SPACE]"
1080 gosub760
1090 printspc(30)m$mt;p$ms;p$
1100 printspc(30)l$lt;p$ls;p$
1110 ifvt=0andvs=0thenprintspc(31)"--[S
     PACE]--";:goto1130
1120 printspc(30)v$vt;p$vs;p$;
1130 pokecc,31
1140 ifi=0theni=1:goto230
1150 return
1160 rem08 zeker weten
1170 pokecc,23:printq$(2)"[3xCRSR-DOWN]
     [CTRL-2]Zeker[SPACE]weten[SPACE]?"
     ;
1180 gett$:ift$<>""thenreturn
1190 goto1180
1200 rem09a mededelingvak wissen
1210 printq$(2)"[2xCRSR-DOWN]":fort=1to
     4:print"[CTRL-2][16xSPACE][16xCRSR
     -LEFT][CRSR-UP]";
1220 next:return
1230 rem09b pos+asc-vak wissen
1240 printq$(2)"[3xCRSR-DOWN]"tab(30);:
     fort=1to4:print"[8xSPACE][8xCRSR-L
     EFT][CRSR-UP]";:next
1250 printq$(0)"[4xCRSR-RIGHT][5xSPACE]
     ":print"[CRSR-DOWN][4xCRSR-RIGHT][
     7xSPACE]":return
1260 rem10verander waarde
1270 printq$(1)"[4xCRSR-RIGHT][3xSPACE]
     [3xCRSR-RIGHT][SPACE]";:pokef,80
1280 z=1788:k=55:pokex,k:pokey,201:op=p
     :gosub1820:p=op:gosub530
1290 pokes+fna(40),pb:op=p:p=p+1:gosub5
     30:pokef,13:return
1300 rem11a tekst type mode
1310 printq$(0)tab(14)"[CTRL-2]Tx":prin
     tq$(2)"Tekst[SPACE]stand[5xSPACE]"
     :print"[6xCOM-`]"
1320 print"[CTRL-9]RETURN[CTRL-0][SPACE
     ]=[SPACE]Terug":print"[CTRL-9][4xS
     PACE]F7[CTRL-0][SPACE]=[SPACE]Rege
     l[2xSPACE]";:gosub1250:b$="[CTRL-0
     ]"
1330 gett$:ift$=""ort$="[CRSR-LEFT]"ort
     $="[CRSR-RIGHT]"ort$="[CRSR-DOWN]"
     ort$="[CRSR-UP]"then1330
1340 p=fna(16):a=asc(t$):v=int(p/16):po
     ke211,p-16*v:poke214,v:sys58732
1350 ifa=18thenb$="[CTRL-9]"
1360 ifa=146thenb$="[CTRL-0]"
1370 ifa=13thengosub1210:printq$(0)tab(
     14)"[2xSPACE]":return
1380 ifa=136thengosub1440:goto1310
1390 ifa<32or(a>127anda<160)then1330
1400 printb$t$
1410 op=p:p=p+1:gosub530
1420 goto1330
1430 rem11b regelmode
1440 gosub1210:pokef,80:op=p:pokex,23:p
     okey,217:n$=""
1450 ifir=1thenprintq$(2)"[2xCRSR-DOWN]
     [2xCOM-`][CRSR-DOWN][2xCRSR-LEFT][
     CTRL-9]F4[CTRL-0][SPACE]=[SPACE]vo
     rige[SPACE]reg";
1460 printq$(2);:fort=1to16
1470 gett$:ift$=""then1470
1480 a=asc(t$):ifa=13then1550
1490 ifa=18thenb$="[CTRL-9]"
1500 ifa=146thenb$="[CTRL-0]"
1510 ifa=20andt>1thenpokex,peek(x)-8:pr
     into$;:t=t-1:n$=left$(n$,2*t-2):go
     to1470
1520 ifa=138andir=1thenn$=ot$:printq$(2
     )n$:goto1550
1530 ifa<32or(a>127anda<160)then1470
1540 n$=n$+b$+t$:printb$t$;:pokex,23+8*
     t:next
1550 ot$=n$:ir=1:pokef,13:p=op:gosub530
     :printq$(2)"[CRSR-DOWN][CTRL-0][6x
     COM-`]"
1560 print"[CTRL-9]RETURN[CTRL-0][SPACE
     ]=[SPACE]Terug":print"[CTRL-9][SPA
     CE]SPACE[CTRL-0][SPACE]=[SPACE]Pri
```

```
     nten";
1570 gett$:ift$=""then1570
1580 p=fna(16):a=asc(t$):ifa=13thenop=p
     :return
1590 ifa=19thenp=0:gosub530
1600 ifa=32then1620
1610 goto1570
1620 poke214,((peek(y)-49)/8):q=((peek(
     x)-23)/8):poke211,q:sys58732
1630 printleft$(n$,32-2*q):l=len(n$)/2:
     ifp+l<255thenprintmid$(n$,32-2*q+1
     ,l*2)
1640 pokef,80:op=p:p=p+1:gosub530:pokef
     ,13:poke198,0:goto1570
1650 rem12 vul sector
1660 pokef,80:printq$(2)"[CTRL-2]Vul[SP
     ACE]sector":print"[CRSR-DOWN]waard
     e[SPACE]:[SPACE]";:k=95:pokex,k
1670 pokey,233:z=1953
1680 gosub1820:gosub1210
1690 poke49155,pb:sys49152:n=1:gosub197
     0
1700 p=0:gosub530
1710 pokecc,31:return
1720 rem13 zoekfunctie
1730 printq$(2)"[CTRL-2]Zoekfunctie":pr
     int"Geef[SPACE]waarde":print"tusse
     n";:k=95:pokex,k
1740 pokey,233:z=1953:pokef,80
1750 gosub1820:p1=pb
1760 iftt$=""thenpoke250,0:si=0:poke492
     61,1:poke1716,32:goto1790
1770 print:print"[4xSPACE]en";:k=95:pok
     ex,k:pokey,241:z=1993:gosub1820
1780 poke250,1:si=1:poke1716,90:poke492
     61,15
1790 poke49229,p1:poke49233,pb:gosub194
     0
1800 gosub530:pokecc,31:pokef,13:gosub1
     210:return
1810 rem14 getal-invoerroutine
1820 pb=0:tt$="":fort=0to2:pokex,k+8*t:
     poke198,0
1830 gett$:ift$=""then1830
1840 ift$="0"then1870
1850 ifasc(t$)=13then1900
1860 ifval(t$)=0then1830
1870 tt$=tt$+t$:pokez+t,asc(t$):next
1880 pokex,k+24:ifval(tt$)>255thenfort=
     0to2:pokez+t,32:next:goto1820
1890 gett$:ift$<>chr$(13)then1890
1900 pb=val(tt$)
1910 pokef,13:return
1920 rem15 schermroutine aanpassen
1930 print#15,"b-p";5;0:print#15,"m-w";
     s$;s$;s$;chr$(65)
1940 pokee+33,177:pokee+34,252:pokee+35
     ,157:pokee+36,0:pokee+37,193+n
1950 syse:pokee+33,189:pokee+34,0:pokee
     +35,193:pokee+36,145:pokee+37,252
1960 return
1970 pokee+20,193+n:pokee+35,193+n:syse
     :pokee+20,193:pokee+35,193:return
1980 rem16 eerste instelroutine
1990 gett$:ift$=""then1990
2000 gosub680:ift$=chr$(135)thenmt=it:m
     s=is:gosub1210:goto810
2010 ift$="e"thengosub470
2020 ift$="[SHIFT-CLR]"thenit=18:is=0:g
     osub760
2030 goto1990
2040 rem17 data inlezen
2050 ifpeek(49289)=96andpeek(53167)=11t
     hen2080
2060 fort=0to137:reada:c=c+a:poke49152+
     t,a:next:pokeb,14:ifc<>20467then21
     90
2070 fort=0to175:reada:d=d+a:poke52992+
     t,a:next:pokeb,8:ifd<>21111then220
     0
2080 ifpeek(860)=192then2130
2090 fort=832to896:poket,0:next
2100 fort=832to858step3:poket,128:next
2110 fort=833to858step3:poket,64:next
2120 poke832,255:poke833,192:poke859,25
     5:poke860,192
2130 dimq$(2):fort=0to2:q$(t)="[HOME][C
     RSR-DOWN]":forg=1to16+2*tstep2:q$(
     t)=q$(t)+"[2xCRSR-DOWN]":nextg,t
2140 pokeb,2:dimn(36):fort=1to35:n(t)=2
     0-w:ift>16thenw=2
2150 ift>23thenw=3
2160 ift>29thenw=4
2170 next:n(36)=20
2180 pokeb,6:return
2190 print"Fout[SPACE]in[SPACE]DATArege
     ls[SPACE]deel[SPACE]1":goto2210
2200 print"Fout[SPACE]in[SPACE]DATArege
     ls[SPACE]deel[SPACE]2"
2210 poke808,237:poke53265,27:end
2220 rem18 scherm printen
2230 print"[HOME][CTRL-2][2xSPACE][11xC
     OM `][3xCRSR-RIGHT][COM-8][SHIFT--
     ][4xCOM-`]"
2240 print"[CTRL-2][2xSPACE][CTRL-9]DIS
     K[SPACE]DRIVER[CTRL-0][3xSPACE][CO
     M 8][SHIFT--][CTRL-9]CRSR[CTRL-0][
     SPACE]Cursor[SPACE]besturing"
2250 printtab(16)"[SHIFT--][CTRL-9]HOME
     [CTRL-0][SPACE]Cursor[SPACE]pos[SP
     ACE]0"
2260 printtab(16)"[SHIFT--][CTRL-9]shZ[
     SPACE][CTRL-0][SPACE]Instel[SPACE]
     +Track"
2270 printtab(16)"[SHIFT--][CTRL-9]Z[3x
     SPACE][CTRL-0][SPACE]Instel[SPACE]
     -Track"
2280 printtab(16)"[SHIFT--][CTRL-9]shX[
     SPACE][CTRL-0][SPACE]Instel[SPACE]
     +Sector"
2290 printtab(16)"[SHIFT--][CTRL-9]X[3x
     SPACE][CTRL-0][SPACE]Instel[SPACE]
     -Sector"
2300 printtab(16)"[SHIFT--][CTRL-9]CLR[
     SPACE][CTRL-0][SPACE]Instel[SPACE]
     18[2xSPACE]0"
2310 printtab(16)"[SHIFT--][CTRL-9]F5[2
     xSPACE][CTRL-0][SPACE]Instel[SPACE
     ]lezen"
2320 print"[CTRL-2]Joris[SPACE]vd[SPACE
     ]Heuvel[SPACE][COM-8][SHIFT--][CTR
     L 9]>[3xSPACE][CTRL-0][SPACE]Volge
     nde[SPACE]sector"
2330 printtab(16)"[SHIFT--][CTRL-9]<[3x
     SPACE][CTRL-0][SPACE]Vorige[SPACE]
     sector"
```

```
2340 printtab(16)"[SHIFT--][CTRL-9]L[3x
     SPACE][CTRL-0][SPACE]Volg[SPACE]li
     nk"
2350 print"[CTRL-2][5xSPACE]1989[7xSPAC
     E][COM-8][SHIFT--][CTRL-9]K[3xSPAC
     E][CTRL-0][SPACE]Vorige[SPACE]Tr/S
     r"
2360 printtab(16)"[SHIFT--][CTRL-9]S[3x
     SPACE][CTRL-0][SPACE]Schrijf[SPACE
     ]sector"
2370 printtab(16)"[SHIFT--][CTRL-9]T[3x
     SPACE][CTRL-0][SPACE]Tekst[SPACE]s
     tand"
2380 printtab(16)"[SHIFT--][CTRL-9]A[3x
     SPACE][CTRL-0][SPACE]Verander[SPAC
     E]waarde"
2390 printtab(17)"[CTRL-9]V[3xSPACE][CT
     RL 0][SPACE]Vul[SPACE]sector"
2400 printtab(16)"[SHIFT--][CTRL-9]H[3x
     SPACE][CTRL-0][SPACE]Roep[SPACE]Sr
     [SPACE]terug"
2410 printtab(17)"[CTRL-9]*[3xSPACE][CT
     RL 0][SPACE]Zoekfunctie"
2420 printtab(16)"[SHIFT--][CTRL-9]E[3x
     SPACE][CTRL-0][SPACE]Eindigen"
2430 printq$(0)"[CRSR-UP][9xSHIFT-*][CO
     M R][SHIFT-*][COM-R][SHIFT-*][COM-
     R][2xSHIFT-*][COM-W]"
2440 print"Pos[6xSPACE][SHIFT--][CTRL-2
     ][SPACE][COM-8][SHIFT--][CTRL-2][S
     PACE][COM-8][SHIFT--][CTRL-2][2xSP
     ACE][COM-8]"
2450 print"[9xSHIFT-*][COM-E][SHIFT-*][
     SHIFT-+][SHIFT-*][COM-E][2xSHIFT-*
     ][COM-W]"
2460 print"Asc[CTRL-2][8xSPACE][COM-8][
     SHIFT--]"
2470 print"[11xSHIFT-*][COM-E][4xSHIFT-
     *][SHIFT-+][22xSHIFT-*]"
2480 print"[CTRL-2]Maak[SPACE]instellin
     g[SPACE][COM-8][SHIFT--]Instelling
     "
2490 print"[CTRL-2]Druk[SPACE]F5[9xSPAC
     E][COM-8][SHIFT--]Deze[3xSPACE]Tr/
     Sr"
2500 printtab(16)"[COM-8][SHIFT--]Link[
     3xSPACE]Tr/Sr"
2510 printtab(16)"[SHIFT--]Vorige[SPACE
     ]Tr/Sr[CTRL-2][HOME]";
2520 return
2530 rem data deel 1
2540 data162,0,169,128,157,0,194,232
2550 data208,250,96,160,0,162,5,32
2560 data198,255,162,0,32,207,255,157
2570 data0,193,232,208,247,32,204,255
2580 data96,160,0,162,5,32,201,255
2590 data162,0,189,0,195,32,210,255
2600 data232,208,247,32,204,255,96,169
2610 data4,133,253,162,0,134,252,169
2620 data16,133,254,160,0,165,250,240
2630 data15,189,0,193,201,0,144,8
2640 data201,100,176,4,169,1,133,249
2650 data189,0,193,145,252,165,252,133
2660 data247,24,165,253,105,212,133,248
2670 data165,249,145,247,169,1,133,249
2680 data232,200,192,16,208,207,24,165
2690 data252,105,40,133,252,165,253,105
2700 data0,133,253,198,254,165,254,208
2710 data186,96
2720 rem20data deel 2
2730 data120,169,148,141,20,3,169,207
2740 data141,21,3,88,96,166,197,172
2750 data141,2,224,2,240,7,224,7
2760 data240,28,76,80,207,173,6,208
2770 data192,1,240,9,24,105,8,141
2780 data6,208,76,80,207,56,233,8
2790 data141,6,208,76,80,207,173,7
2800 data208,192,1,240,9,24,105,8
2810 data141,7,208,76,80,207,56,233
2820 data8,141,7,208,234,234,234,234
2830 data166,2,189,168,207,232,224,8
2840 data208,2,162,0,141,42,208,134
2850 data2,173,6,208,201,151,240,18
2860 data201,15,240,22,173,7,208,201
2870 data177,240,20,201,41,240,24,76
2880 data49,234,169,23,141,6,208,76
2890 data49,234,169,143,76,124,207,169
2900 data49,141,7,208,76,49,234,169
2910 data169,76,137,207,166,251,224,4
2920 data240,6,232,134,251,76,49,234
2930 data162,0,134,251,76,13,207,234
2940 data0,11,12,15,1,15,12,11
```

** EINDE LISTING disky

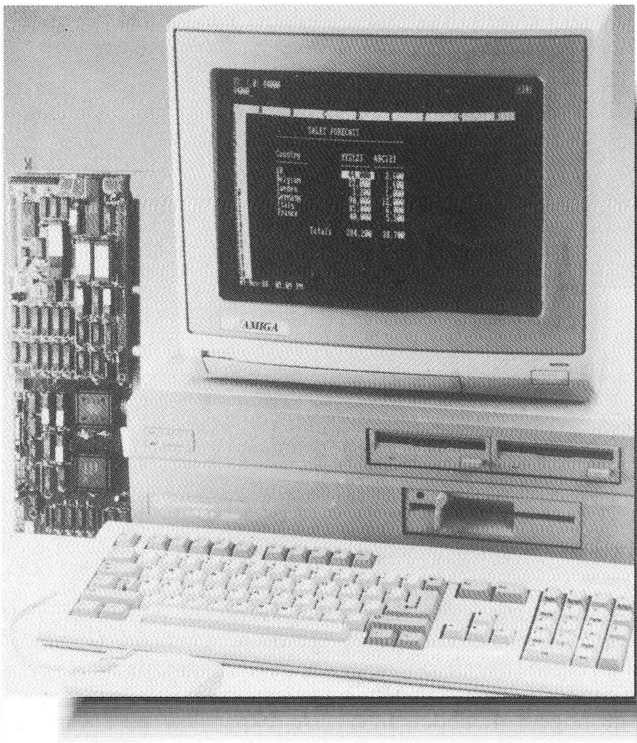
Checksum disky

REGEL 10	159
REGEL 20	155
REGEL 30	227
REGEL 40	217
REGEL 50	58
REGEL 100	185
REGEL 110	9
REGEL 120	63
REGEL 130	59
REGEL 140	155
REGEL 150	122
REGEL 160	163
REGEL 170	203
REGEL 180	252
REGEL 190	93
REGEL 200	218
REGEL 210	16
REGEL 220	19
REGEL 230	61
REGEL 240	38
REGEL 250	92
REGEL 260	130
REGEL 270	160
REGEL 280	156
REGEL 290	123
REGEL 300	75
REGEL 310	117
REGEL 320	149
REGEL 330	130
REGEL 340	82
REGEL 350	179
REGEL 360	221
REGEL 370	102
REGEL 380	97
REGEL 390	88
REGEL 400	168
REGEL 410	62
REGEL 420	35
REGEL 430	121
REGEL 440	77
REGEL 450	52
REGEL 460	53
REGEL 470	240
REGEL 480	62
REGEL 490	37
REGEL 500	29
REGEL 510	108
REGEL 520	112
REGEL 530	112
REGEL 540	45
REGEL 550	107
REGEL 560	112
REGEL 570	34
REGEL 580	114
REGEL 590	187
REGEL 600	132
REGEL 610	117
REGEL 620	246
REGEL 630	12
REGEL 640	118
REGEL 650	213
REGEL 660	73
REGEL 670	183
REGEL 680	192
REGEL 690	194
REGEL 700	61
REGEL 710	190
REGEL 720	104
REGEL 730	233
REGEL 740	57
REGEL 750	59
REGEL 760	132
REGEL 770	99
REGEL 780	207
REGEL 790	142
REGEL 800	125
REGEL 810	159
REGEL 820	12
REGEL 830	97
REGEL 840	121
REGEL 850	6
REGEL 860	199
REGEL 870	119
REGEL 880	53
REGEL 890	62
REGEL 900	156
REGEL 910	141
REGEL 920	202
REGEL 930	134
REGEL 940	230
REGEL 950	152
REGEL 960	139
REGEL 970	95
REGEL 980	84
REGEL 990	35
REGEL 1000	81
REGEL 1010	32
REGEL 1020	172
REGEL 1030	103
REGEL 1040	172
REGEL 1050	102
REGEL 1060	194
REGEL 1070	112
REGEL 1080	42
REGEL 1090	219
REGEL 1100	216
REGEL 1110	127
REGEL 1120	49
REGEL 1130	173
REGEL 1140	225
REGEL 1150	142
REGEL 1160	251
REGEL 1170	243
REGEL 1180	51
REGEL 1190	83
REGEL 1200	194
REGEL 1210	121
REGEL 1220	74
REGEL 1230	22
REGEL 1240	195
REGEL 1250	171
REGEL 1260	251
REGEL 1270	133
REGEL 1280	181
REGEL 1290	253
REGEL 1300	36
REGEL 1310	184
REGEL 1320	123
REGEL 1330	142
REGEL 1340	200
REGEL 1350	252
REGEL 1360	174
REGEL 1370	224
REGEL 1380	157
REGEL 1390	25
REGEL 1400	119
REGEL 1410	103
REGEL 1420	80
REGEL 1430	199
REGEL 1440	35
REGEL 1450	250
REGEL 1460	201
REGEL 1470	191
REGEL 1480	16
REGEL 1490	252
REGEL 1500	174
REGEL 1510	121
REGEL 1520	164
REGEL 1530	30
REGEL 1540	12
REGEL 1550	14
REGEL 1560	100
REGEL 1570	192
REGEL 1580	136
REGEL 1590	32
REGEL 1600	83
REGEL 1610	86
REGEL 1620	103
REGEL 1630	239
REGEL 1640	51
REGEL 1650	185
REGEL 1660	98
REGEL 1670	204
REGEL 1680	227
REGEL 1690	61
REGEL 1700	145
REGEL 1710	117
REGEL 1720	58
REGEL 1730	124
REGEL 1740	119
REGEL 1750	87
REGEL 1760	47
REGEL 1770	94
REGEL 1780	226
REGEL 1790	119
REGEL 1800	6
REGEL 1810	135
REGEL 1820	136
REGEL 1830	191
REGEL 1840	160
REGEL 1850	161
REGEL 1860	110
REGEL 1870	12
REGEL 1880	162
REGEL 1890	175
REGEL 1900	38
REGEL 1910	53
REGEL 1920	119
REGEL 1930	85
REGEL 1940	4
REGEL 1950	41
REGEL 1960	142
REGEL 1970	83
REGEL 1980	179
REGEL 1990	198
REGEL 2000	113
REGEL 2010	13
REGEL 2020	7
REGEL 2030	92
REGEL 2040	38
REGEL 2050	28
REGEL 2060	216
REGEL 2070	161
REGEL 2080	247
REGEL 2090	172
REGEL 2100	241
REGEL 2110	193
REGEL 2120	169
REGEL 2130	13
REGEL 2140	83
REGEL 2150	216
REGEL 2160	223
REGEL 2170	216
REGEL 2180	3
REGEL 2190	225
REGEL 2200	90
REGEL 2210	36
REGEL 2220	218
REGEL 2230	96
REGEL 2240	228
REGEL 2250	58
REGEL 2260	117
REGEL 2270	220
REGEL 2280	206
REGEL 2290	53
REGEL 2300	218
REGEL 2310	89
REGEL 2320	202
REGEL 2330	233
REGEL 2340	67
REGEL 2350	106
REGEL 2360	189
REGEL 2370	234
REGEL 2380	221
REGEL 2390	210
REGEL 2400	187
REGEL 2410	166
REGEL 2420	25
REGEL 2430	237
REGEL 2440	198
REGEL 2450	141
REGEL 2460	177
REGEL 2470	41
REGEL 2480	102
REGEL 2490	45
REGEL 2500	176
REGEL 2510	6
REGEL 2520	142
REGEL 2530	244
REGEL 2540	189
REGEL 2550	81
REGEL 2560	245
REGEL 2570	233
REGEL 2580	79
REGEL 2590	133
REGEL 2600	43
REGEL 2610	182
REGEL 2620	224
REGEL 2630	32
REGEL 2640	180
REGEL 2650	37
REGEL 2660	80
REGEL 2670	48
REGEL 2680	24
REGEL 2690	70
REGEL 2700	38
REGEL 2710	189
REGEL 2720	87
REGEL 2730	235
REGEL 2740	154
REGEL 2750	229
REGEL 2760	144
REGEL 2770	29
REGEL 2780	48
REGEL 2790	93
REGEL 2800	33
REGEL 2810	143
REGEL 2820	186
REGEL 2830	199
REGEL 2840	122
REGEL 2850	125
REGEL 2860	169
REGEL 2870	175
REGEL 2880	149
REGEL 2890	50
REGEL 2900	158
REGEL 2910	249
REGEL 2920	189
REGEL 2930	179
REGEL 2940	110

DE AMIGA-FAMILIE

Volledig overzicht van de Commodore 680X0-lijn

De Commodore Amiga valt niet meer als spelletjesmachine, video-, muziek/geluid- en dtp-computer of alternatieve PC weg te denken. Er is voor elk wat wils. Voor eenvoudige huiselijke toepassingen en games verschaft de Amiga 500 alle faciliteiten. De modellen 2000, 2500 en komende 3000 bieden alle zakelijke mogelijkheden plus UNIX, dtv en dtp. Het oermodel Amiga 1000 is alleen nog in het tweedehands-circuit actief. Daar geldt deze oudgediende als krachtige hobbycomputer waar nog veel aan te verbouwen valt. Om de keuze bij aankoop te vergemakkelijken hebben we de complete Amiga-familie-hardware weer eens op een rijtje gezet.



Amiga 2000

De coprocessors

Bij de Commodore Amiga doet de MC 680X0 CPU het werk niet alleen. Meerdere coprocessors nemen de centrale microprocessor veel werk uit handen waardoor de systeemprestaties en de grafische kwaliteit flink toenemen.
De Amiga-coprocessors staan hieronder vermeld.

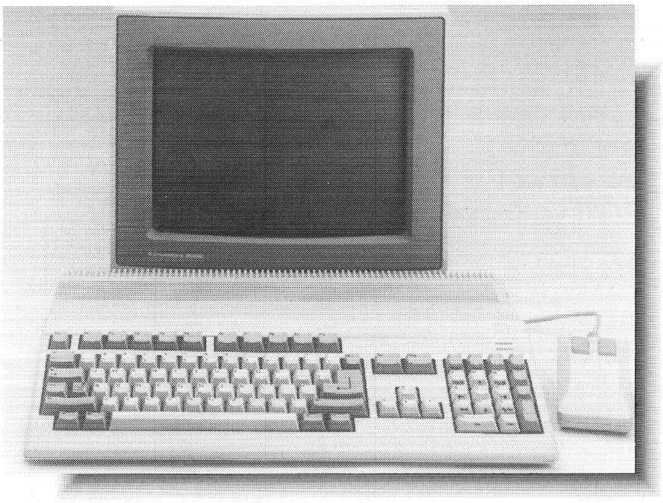
- De videochip Denise kent vier verschillende grafische schermen: 320 x 256, 320 x 512, 640 x 256 en 640 x 512 pixels. Het 50 Hz videobeeld heeft een oplossend vermogen van 625 lijnen en een geheugen van maximaal 512 KB. In de tekstmodus zijn naar keuze 60 of 80 tekens beschikbaar met maximaal 25 gekleurde regels. Het kleurenpalet biedt 32 kleuren (bij 320 beeldlijnen) of 16 kleuren (bij 640 lijnen) met maximaal 4096 mengkleuren. Verder verplaatst Denise de "playfields". In samenwerking met Agnus kan Denise circa 1.000.000 pixels per seconde optekenen en gekleurde vlakken zo snel opvullen dat het menselijk oog dit niet merkt.
- De DMA-chip Paula beheerst de geheugentoegang van de coprocessors. Daarnaast controleert Paula de I/O-besturing van de parallelle en seriële interfaces, de control ports, het uitlezen van het toetsenbord en de audio in/output. Het stereodeel omvat vier toonkanalen met twee stereo-uitgangen, programmeerbare samplingrate, amplitude, 9 octaven en frequentie-modulatie.
- Gate-Array-Chip Gary vervuld snel en nauwkeurig tal van logische functies, o.a. het aansturen van TTL- monitors. Na de Amiga 500 verscheen Gary op alle Amiga-modellen vanaf 2000B.
- De Bit-Blitter- of animatie-chip Agnus. Van deze chip zijn er verschillende typen in omloop die we elk afzonderlijk zullen bespreken.

Bij de introductie van de Amiga 500 was er voor het eerst sprake van een speciale custom-uitvoering, de zogenaamde Fat Agnus, waarbij nog meer functies in een en dezelfde IC zijn ondergebracht. Het verschil met de oude Agnus zit hem in de hoeveelheid RAM die de chip direct kan aangrijpen. De dunne kon slechts 512 KB, de dikke kan rechtstreeks 1 MB Chip-RAM aan.
Momenteel zit de Fat Agnus 8371 in de meeste uitgebrachte Amiga-modellen. Deze coprocessor verzorgt in nauwe samenwerking met de beide anderen de vloeiende Amiga-animaties.
Ook Fat Agnus, de DMA-adresgenerator, is onmisbaar voor het creëren van de flitsende Amiga-animaties. Deze dikzak kan in de Amiga bliksemsnel 1MB aan videodata verversen. Direct Memory Access omzeilt de hoofdprocessor en spreekt het geheugen dus direct aan. De DMA-controller hebben wij zojuist al besproken. Het belangrijkste onderdeel van Agnus is echter de coprocessor Copper. Copper wordt in feite geheel door de elektro-

nenbundel van de monitor gestuurd en schrijft de data van en naar de registers. Er zijn drie bevelen mogelijk: MOVE verplaatst de data, WAIT doet de coprocessor wachten tot de elektronenbundel de gewenste schermplaats bereikt heeft en SKIP voor het overslaan van bevelen. Copper verandert slechts de registers en grijpt niet zelf op het RAM in.
Het derde IC-onderdeel van Agnes is de zogenaamde Blitter. De Blitter is een soort grootschalige verhuizer van grafische datablokken. Commodore zelf geeft 1.000.000 pixels en 60 beelden per seconde op. Objecten die groter zijn dan een sprite of het vullen van kleurvlakken verlopen via de Blitter in een mum van tijd. Interessanter is echter de animatie-optie om maximaal drie bewegende elementen met elkaar te kunnen verbinden. Bijvoorbeeld de benen en het hoofd van een cartoon- figuurtje animeren.
Sinds kort is er ook weer een nieuwere versie van de Fat Agnus, nu Big Agnus 8372 uitgekomen.

De interfaces

Gelukkig heeft Commodore bij de Amiga 500, 2000/2500 en 3000 de eigenzinnige interface-politiek laten varen. Alleen de niet meer in produktie zijnde Amiga 1000 heeft nog een afwijkende parallelle printerpoort. **Gebruik daarvoor uitsluitend de speciale Amiga 1000-printerkabel anders brandt de boel door!**
Bij de overige Amiga's zijn de poortaansluitingen dus allemaal op PC-standaard. De RSC 232-uitgang laat zich moeiteloos aansluiten op tal van modems, afdrukkers en communicatie-kabeltjes. Alleen voor MIDI-toepassingen is een speciale adapterkabel nodig. De Baudrate is programmeerbaar tot maximaal 31.250 baud en het transport van ASCII-files mag met geschikte communicatie-software dus geen problemen meer opleveren. Datzelfde geldt eveneens voor de parallelle Centronics-poort. Een speciale adapterkabel is niet meer nodig.
Een tweede parallelle interface verzorgt de communicatie met ten hoogste drie externe drives. Er is keuze uit 3.5- en 5.25-inch floppy drives en een harddisk met ingebouwde controller van Commodore zelf of onafhankelijke merken. De Amiga 500 kan maximaal 1 extern floppystation voeden. Meerdere floppy drives of een harddisk zullen hun eigen stroomvoorziening moeten meebrengen! Bij de hogere modellen ligt de voeding wat ruimer.
Een monitor kan zowel via een tulpkabel (BAS-monochroom, bij de 2000/2500 FBAS-monochroom) als een 25-pins RGB A/D-plug aangesloten worden. Wie nog geen geld voor een kleurenmonitor heeft kan dus nog vooruit met zijn of haar oude monochrome model. Bezitters van een A1081 kunnen al direct van de fraai gekleurde HIRES-graphics genieten.
Op de meeste 500-modellen ontbreekt een RF-modulator. Deze zijn natuurlijk ook los te koop, maar dat betekent weer een extra uitgave, een ontsierend kastje op je bureau en een verminderde beeldkwaliteit.
De geluidsprestaties van de Amiga-familie komen via een interne luidspreker nimmer tot hun recht. Aansluiten met tulpkabeltjes op de huiskamer-stereo of eventueel stereo-tv brengt daarentegen menige moderne muziekliefhebber in verrukking.
De 9-pins muis/joystickpoorten zitten bij de Amiga 500 aan de achterzijde, bij de 1000 aan de rechterzijde en bij de hogere modellen aan de voorzijde. Het een is wat minder handig dan het andere. Gelukkig passen er wel gewoon standaard Commodore/Atari- compatibele spelpoken op.
De expansion port van de modellen 500 en 1000, in feite een processorbus, biedt de veeleisende gebruiker nog de mogelijkheid om meer RAM-geheugen, een extra coprocessor, een harddisk, meet/regel-relais, robot-controllers of andere randapparatuur te installeren. Vreemd is dat bij de eigen MS-DOS-emulator Sidecar op de 512 een aansluitprobleem dreigt te ontstaan. De systeembus zit aan de verkeerde kant en is nog 180 graden gedraaid ook. Een adapterkabeltje brengt uitkomst, maar het geheel wordt er niet fraaier op. Bij de 1000 geeft de Sidecar (alleen nog tweedehands leverbaar) geen probleem.


Amiga 500

Het toetsenbord

Het toetsenbord is, behalve bij het model 1000, van het AT-type. De enige afwijking zit hem in de A-toetsen (1000, 2000/2500) en CBM-toets (Amiga 500). Bij de Amiga 500 is het keyboard in de systeemkast geïntegreerd. De overige toetsenborden zijn met een spiraalkabeltje aan de Amiga-systeemkast verbonden.
De layout oogt modern en ziet er met 96 toetsen professioneel uit.

De muis

Alle Amiga's hebben een staartje met een muis er aan. Het digitale Amiga-knaagdier is een eenvoudig opto-mechanisch model met twee bedieningsknoppen. Een kunststoffen bal brengt de bewegingen op het bureau aan de sensors over. Voor het noodzakelijke reinigen zit er aan de onderzijde een handig klapluikje.
De muis is tamelijk fors en vergt behoorlijk wat tafel- en elleboogruimte. Na even wennen wordt de 500-muis echter tot een betrouwbaar een plezierig aanwijsgereedschap. Voor games en het maken van tekeningen prefereert de redactie echter respectievelijk een joystick en een lichtpen.

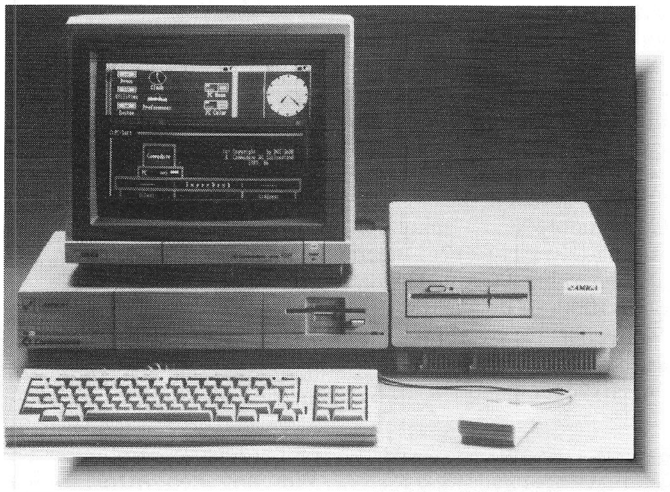
Het RAM

Bij de Amiga 1000 is het RAM beperkt tot 256 Kb op het moederbord en 256 KB via de uitbreidingskaart aan de voorzijde van de systeemkast. Gelukkig zag een aantal onafhankelijke leveranciers dat deze 512 KB toch wel wat te beperkt is. Momenteel zijn er Amiga 1000 RAM-uitbreidingen tot 8 MB leverbaar. Voor de overige modellen kunt u met een welgevulde beurs tot 16 MB uitbreiden.
De Amiga 500 en 2000/2500 en 3000 kennen twee verschillende soorten RAMs. Memory boards die de 680X0-bus gebruiken bieden

het zogenaamde fast memory. Fast wil zeggen dat het RAM de bus-access niet met de video Blitter-chip behoeft te delen. Dat was wel het geval op de Amiga 1000 waarbij de Blitter, dankzij zijn hogere systeemprioriteit, schaamteloos buscycli stal van het interne chip memory hetgeen de programma-uitvoer vertraagde.
Het verdient altijd aanbeveling meer MB in uw Amiga te zetten, dat verbetert en versnelt de prestaties aanmerkelijk. Vanaf het model 2500 stopt Commodore van huis uit al meerdere MB's via een RAM-expansion-kaart in de systeemkast.

De Amiga 500

In tegenstelling tot de Amiga 2000 is de in de VS ontwikkelde Amiga 500 geen bouwdoos (een open-systeemarchitectuur), maar een vrijwel gesloten compleet systeem. In de op de C-128 lijkende systeemkast met ingebouwde 3.5-inch diskdrive is geen ruimte voor meerdere inbouwkaarten of extra drives. Hoogstens kan er aan de onderzijde nog een geheugenuitbreidingskaart met klok ingestoken worden. Alle verdere uitbreidingen moeten via de expansion port aan de linkerzijde van de computer.
De Amiga 500 biedt de serieuze hobbyist meteen alles wat voor jarenlang probleemloos computeren en doorgroeien noodzakelijk is. Het ingebouwde 880 KB 3.5-inch diskettestation kan qua datacapaciteit de meeste spelen en programmatuur ruimschoots aan. Wie meer mogelijkheden of gebruikersgemak ambieert kan via de floppy-poort nog een extra drive aansluiten.
Standaard zijn een 256KB metend ROM met het operating system Kickstart 1.2 of 1.3, en een 512KB groot RAM aan boord. Desgewenst kan de geheugencapaciteit tot meerdere MegaBytes opgevoerd worden. Deze uitbreiding kan bovendien nog in combinatie met een PC-kaart of turbokaart. Kortom, ruimte genoeg dus voor grote spreadsheet-modellen, gedetailleerde graphics of videobeelden en semi- zakelijk PC-gebruik (onder MS-DOS).
Het hart van de Amiga 500 wordt net als bij de 1000 en 2000 gevormd door de 32/16-bits Motorola MC 68000 CPU met een kloksnelheid van 7,16 MHz. De interne (adresbus) data-overdracht is 32-bits. Data-overdracht per systeembus verloopt in woorden van 16 bit. Verder beschikt de MC 68000 over een aanzienlijk krachtiger on board instructieset dan de 6502 CPU's van de C-64 en C-128. Dat bespaart veel tijd bij het programmeren en het uitvoeren van opdrachten. Bovendien behoort multi-tasking tot de mogelijkheden.
De drie coprocessors zijn momenteel alleen in de modernste versie leverbaar.
In de Amiga 500 vinden we links aan de voorzijde de standaard geheugenbank met 512KB aan vrij RAM. Deze bank bestaat uit 16 256KB RAM-chips met een accestijd van 150 nanoseconden. Aan de rechterzijde van deze RAM- bank kan de uitbreidingskaart met 512KB extra RAM, klok en accu in de daarvoor bestemde connector gestoken worden. Deze ruimte is via de bodemplaat toegankelijk zodat de gebruiker de 500 niet behoeft open te schroeven..
Verder zitten er nog een quarts video-oscillator met een frequentie van 28.63 MHz (dicteert de 68000 een kloksnelheid van 7.16 MHz) en enkele DMA I/O-chips op de hoofdkaart. Veel meer IC's zijn er niet. Kortom een keurig staaltje van geïntegreerde chiptechniek.
De ingebouwde 3.5-inch floppy disk drive biedt een comfortabele 880KB aan geformatteerde opslagcapaciteit. Dat is altijd nog 160KB meer dan de in concurrent Atari ST gebouwde drive. Daar de interne diskdrive zijn voeding uit de Amiga 500 trekt konden de afmetingen bescheiden gehouden worden. Onze drive functioneerde zonder mankeren (een verademing na al die 1541-kuren) en viel zelfs redelijk snel te noemen.


Amiga 1000 met Sidecar

De Amiga 1000

Eigenlijk is deze oer-Amiga sinds 1987 al niet meer leverbaar. Bij de introductie in 1985 gold de Amiga 1000 als The State Of The Art grafische computer. Datgene wat de Motorola MC 68000 met het roemruchte chiptrio op de monitor presteerde sloeg alles wat machines die een veelvoud van die *f* 7.000,- kostten konden.
De tijd gaat echter razend snel. Slechts één 3.5-inch diskdrive aan boord en het beperkte RAM werden snel achterhaald. De mogelijkheden groeiden zelfs in een rap tempo boven de hardware uit. Ondanks de dalende prijs kon de Amiga 1000 niet meer tegen het Atari- en IBM PC-kloon-geweld opboksen.
Voor de hobbyist is er gelukkig nog een grote keuze uit optionele hardware-uitbreidingen. Variërend van harddisks en meer RAM tot professionele genlocks. Hier en daar kunt u wellicht nog de uitbreiding tot IBM-compatibele PC, de Sidecar, opduiken.
Momenteel komt de Amiga 1000 eigenlijk alleen nog maar als veelzijdige hobbymachine in aanmerking. Ideaal voor de doe-het- zelver die een ietsje meer wil.

De Amiga 2000

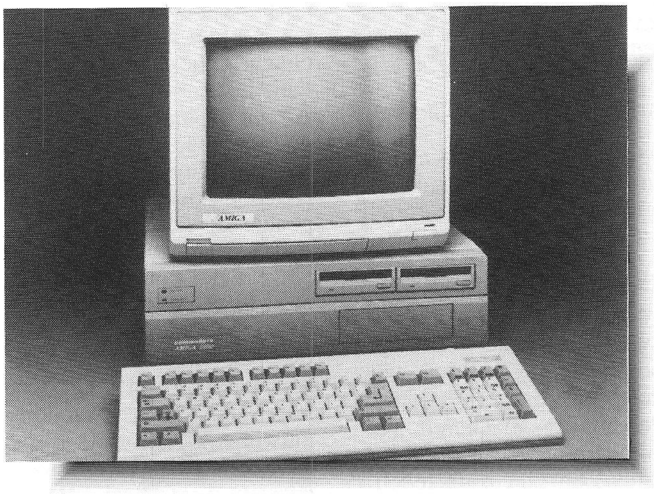
Op de keper beschouwd werd de Amiga 200O puur om concurrentieredenen in Duitsland ontwikkeld. De Amiga-droom leek het tegen de groeiende concurrentie te moeten afleggen en wie een serieuze gooi naar een hoge positie op de overvolle PC-markt wilde doen moest van goede huize komen. Er waren al zo veel voor elk wat wils XTs en ATs te koop dat eentje meer of minder geen opzien baarde. Een voldongen feit waar de Commodore-ontwerpers zich terdege van bewust waren. Zij besloten daarom de grafische, muzikale en rekenkracht van de Amiga 1000 onder Amiga DOS te combineren met het enorme MS-DOS-potentieel door een hybride computer met hardware voor beide besturingssystemen in één huis te ontwerpen. Het resultaat was de Amiga 2000A met de volgende sterke troeven.

- Een Open Systeem Architectuur (OSA) maakt uitbreiding via de zeven ingebouwde expansion

slots mogelijk. Ook zijn er nog twee extra slots voor een 68020 CPU met memory management en een floating-point chip, en een nieuwe PAL-composiet-videokaart.

- Een XT-(Janus Card) en een AT-kaart die afzonderlijk of onder multi-tasking (Amiga DOS of UNIX 5.2) in de Amiga 2000 kan draaien. Via dubbele slotconnectors maakt de kaart aan de ene kant contact met de Amiga- en aan de andere kant met de IBM-PC-hardware (de "ingebouwde Sidecar").
- Het gebruik van Hard Cards/disks die zowel Amiga- als MS-DOS-partitions kunnen bevatten. M.a.w. op een en dezelfde on board harde schijf kan het disk operating system zowel Amiga- als MS-DOS-files wegschrijven en lezen.
- Kickstart zit bij de 2000 in ROM hetgeen het booten aanzienlijk versnelt.
- Standaard beschikte de Amiga 2000 eerst over 512 KB en al snel over 1 Megabyte aan RAM. Maximaal kan 8 MB aan vrij RAM geadresseerd worden.
- Een on board klokkalender met batterij-voeding.

Al deze extra hardware maakt de Amiga 2000 tot een waar Power House, maar niet tot een geheel nieuwe machine. De 7.14 MHz 68000 CPU en de DMA, display- en geluidchips bleven immers hetzelfde. Beter is het om van een sterk opgepepte 1000-versie te spreken.
Commodore's zakelijke Amiga-telg ziet er veel meer als een echte PC uit dan de oude 1000- en de iets jongere 500-versie. De systeemkast is deels van metaal en flink hoger en dieper dan bij de 1000.



Amiga 2000

- Het speciale video-slot maakt HIRES (640 x 512) non- interlaced composietvideo en andere VGA-toepassingen mogelijk.
- Er is nu ruimte voor drie, twee 3.5-inch en één 5.25-inch, floppydrives in de systeemkast.

De 1084 kleurenmonitor lijkt nu niet meer op een soort "waterhoofd" maar past qua grootte goed bij het metalen Amiga- huis. Het verhoogde voorfront biedt plaats aan twee 3.5-inch en een 5.25-inch drive. Verder verhuisden de muis- en de joystick-aansluitingen naar de voorzijde.

Na het openen van de systeemkast worden de negen (er zijn zeven uitbreidingsslots, maar twee daarvan vormen via dubbele connectors een brug tussen de Amiga- en PC-bus!) connector-slots zichtbaar. Een zee aan uitbreidingsmogelijkheden in vergelijking met de enkelvoudige externe connector van de 1000. Vijf van de slots gebruiken de Amiga-systeembus en vier slots gebruiken een speciale secundaire IBM-PC XT-compatibele bus. Middels twee 36- pins connectors is ook al op de aangekondigde AT-kaart gerekend.
Met de zeven uitbreidingsslots zijn diverse configuraties mogelijk. Een aantrekkelijke optie is het gebruik van de A2088 XT PC-emulator (en later eventueel de AT-emulator) in een van de brugslots. Dan blijven er nog drie slots voor PC-kaarten, bijvoorbeeld een VGA- of modem-kaart en een hardcard, en drie voor Amiga-kaarten, bijvoorbeeld een RAM-kaart of harddisk-controller, over. Wie nog een extra Amiga kaartsleuf wil schuift de PC-kaart gewoon een slot naar links en houdt dan vier Amiga- en twee PC-slots over.
Naast de gewone uitbreidingsslots zijn er nog twee speciale Amiga-slots:

- een 86-pins Amiga bus-slot voor de 68000 CPU: hierin kan bijvoorbeeld een 68020 mathematische co-processor met floating point en eigen memory management gestoken worden;
- een speciaal PAL Composiet Video Board-slot.

Let bij aankoop van een Amiga uit dubieuze of tweedehands bron even op of het om een Model A of Model B gaat. Het model B is een soort upgrade van een A met de nieuwe 500-chips. Er zitten een Fat of Big Agnus, Gary, betere buscontroller en PAL- videoprestaties in.

De 2000-bus

Bij de Amiga 2000 is sprake van maar liefst drie bussen. Deze kleine busonderneming valt uiteen in drie soorten.

- De 68000 32-bits CPU-bus met 86 adreslijnen. Het gaat duidelijk om een soort snelbus, want er zit geen enkele buffer tussen het CPU-slot en de Motorola 68000-microprocessor.
- De Amiga-bus met 100 adreslijnen voor CPU-, externe interrupt- en busverkeersregel-signalen. Alle lijnen zijn gebufferd wat het construeren van de interfaces voor de uitbreidingskaarten aanzienlijk vergemakkelijkt. Jammer genoeg is deze busstructuur niet compatibel met de oude Zorro-buskaart voor de Amiga 1000.
- De PC-bus is naar keuze de bekende 62-lijns XT-bus of de 16-bits AT-bus.

Om al het busverkeer in goede banen te leiden heeft de Amiga een eigen perron-opzichter in de vorm van AutoConfig. Het AutoConfig-protocol versie 1.2 en hoger zorgt ervoor dat alle kaarten probleemloos op de bus stappen. Hiervoor zitten op de Amiga-bus de lijnen CONFIG-IN en CONFIG-OUT.
Bij het opstarten of na een RESET staan alle uitbreidingskaarten als het ware op de bus te wachten, de ongeconfigureerde status. CONFIG-IN checkt de eigenschappen (signature data) van de eerste kaart en geeft deze aan de CPU door. Vervolgens configureert de Kickstart-software de kaart aan de hand van de opgegeven adresruimte, initialisatiecode en geheugengebruik. Is het board aldus geconfigureerd dan zendt het een CONFIG-OUT-signaal uit en begint het proces met de daaropvolgende kaart weer van voren af aan. Bij de 2000 verloopt AutoConfig geheel

automatisch via het 256K gemeten on board Kickstart-ROM.

De PC-kaarten

Inmiddels zijn zowel XT- als AT- en 386SX-kaarten voor de Amiga 2000, 25000 leverbaar. Voor de Amiga 500 is er vooralsnog alleen een XT-kaart verkrijgbaar.
Het A2088-board, ook wel Bridgeboard of Janus-kaart genoemd, is in feite een zelfstandige XT-hoofdkaart. Een Sidecar op een kaart dus. De 8088-microprocessor draait onder de standaard 4.77 Mhz klokcyclus en een 16KB EPROM bevat alle noodzakelijke BIOS- routines. Een onlangs door Commodore nieuw uitgebrachte A2088-kaart bevat de tot op maximaal 9,4 MHz draaiende Nec V20 CPU voor de XT-functies. Het BIOS is door Commodore behoorlijk aangepast om communicatie met de Amiga 68000 mogelijk te maken.
De A2088 heeft standaard 512 KB aan boord plus nog eens een extra 128KB voor het dual-port RAM. Een SMC 9268 floppy-controller-chip en een standaard 5.25-inch floppy driveconnector kunnen maximaal twee (een interne en een externe drive) daisy chained floppystations aansturen.
Om alle XT-functies op een enkele kaart te kunnen proppen moest Commodore eigen (custom-)chips ontwerpen. Een speciale multifunction chip functioneert daarbij zowel als een DMA- en interrupt controller als een timer voor alle bussignalen. Andere chips "bedriegen" het Sidecar-deel van de Amiga 2000. De machine "gelooft" dat hij een echte XT is en via het dual-port memory worden alle toetsaanslagen en I/O-opdrachten van de Amiga als zijnde een PC XT geëmuleerd.
Verder biedt de XT-kaart een extra socket voor de 8087 mathematische coprocessor.

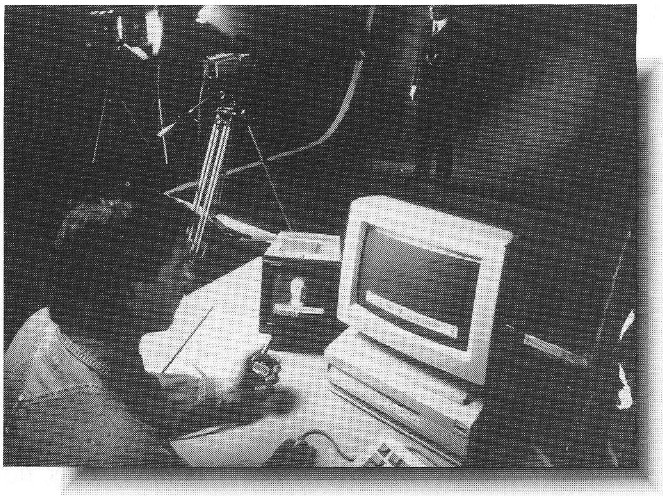
De A2286 Bridgeboard maakt van de Amiga 2000 een volwaardige 8 MHz AT met heuze 1.2 MB diskdrive. Wie meer wil kan in Duitsland al een SX-Bridgeboard aanschaffen.

Amiga 2500

In feite is een Amiga 2500 niets anders dan een model 2000 in professionele uitvoering. Een 2 MB geheugenuitbreiding (A2058) completeert het 1 MB Chip Memory en minimaal een 40 MB harddisk zit er al standaard in. Nog meer rekenkracht komt van de Motorola 68020 CPU op de A2620-Turbokaart. Op deze kaart is tevens nog een vrij socket voor de mathematische coprocessor MC 6881 voorhanden.
De harddisk-controllerkaart is de 2090A of diens opvolger. Deze controller beschikt zowel over een ST506- als een SCSI-interface. Sommige leveranciers hebben het beeldflikkerprobleem bij de hogere schermresoluties maar meteen met de bekende Flickerfixer (704 x 510 pixels) ondervangen.
Verder is deze machine in principe UNIX-compatibel en krijgt u een fraai 2500-logo op de systeemkast. Helaas is deze turbo- uitvoering (op 14,3 MHz ruim twee maal zo snel) van Amiga aan de prijzige kant. Soms loont het zelfs de moeite om gewoon een turbokaart van een onafhankelijk merk in de 2000-systeemkast te steken.

Amiga 3000

Super-Amiga's bestaan eigenlijk al sinds de MC 68030 op de markt verscheen. Met high-tech-dtp en -dtv of wetenschappelijke beeldanalyse werkende gebruikers verbouwden hun Amiga tot een waar power-house. Behalve die 68030 zaten er natuurlijk een 68881/2 mathematische coprocessor, vele MegaBytes aan snel RAM, een HIRES-videokaart en een zware harddisk in de 2000-kast.



Amiga 3000

Commodore zelf heeft nu ook ingezien dat er een markt voor zware Amiga's bestaat. Met de introductie van het model 3000 wordt de wereld der animaties, professionele desktopvideo en desktoppublishing, beeldanalyse, digitale muziek en grafisch ontwerpen met geavanceerde technieken zoals raytracing op haar wenken bediend. Extra is nog de uitbreiding naar interactieve videotechnieken (ITV) via de on board auteurstaal AMIGAVISION.
Wat krijgt de Amiga-professional nu eigenlijk voor rond de *f* 10.000,- in huis? In de eerste plaats een multi-tasking videografisch werkstation. Ideaal voor de toepassingsgebieden multi-media, dtv, dtp, grafisch ontwerpen, CAD/CAM, beeldanalyse, musici (4 kanalen stereo) en wetenschappelijk onderzoek.
Tot de specificaties behoren onder andere:

- een Motorola 68030 CPU met 68882 (op 25 MHz) of 68881 (op 16 MHz) mathematische coprocessor;
- enhanced DMA-chipset, 32-bits busstructuur;
- 1 MB grafisch geheugen, maximaal 2 MB op de hoofdkaart;
- 1 MB Fast-RAM tot maximaal 16 MB uitbreidbaar;
- apart VCE/CACHE-geheugenslot;
- - video PAL (15,75 kHz) of VGA (31,5 kHz); de display-enhancer scoort maximaal 1280 x 512 pixels bij 4096 HAM-kleuren;
- 14-inch 1950 Multisync kleurenmonitor (eindelijk weg met de flikkering!);
- harddisk naar keuze 19 ms type 40 of 100 SCSI;
- besturingssystemen Kickstart 2.0, Workbench 2.0, Amiga Vision; MS-DOS optioneel bij de Bridgeboards.

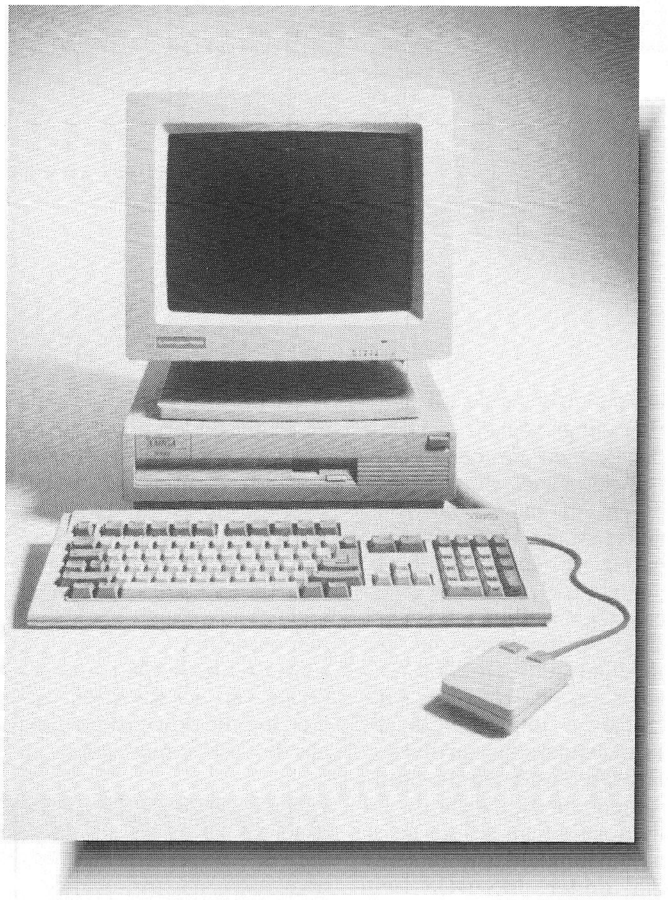
Het systeem is voorlopig in drie oplopende modellen (vanaf *f*8.700,- exclusief BTW) leverbaar. Ons inziens vormt de Amiga 3000 een welkome professionele aanvulling op de reeds 1.500.000 verkochte lagere modellen. In deze prijsklasse zal het moeilijk zijn om een gelijkwaardige grafische, animatie en tevens muzikale computer te vinden.

U.S.

DE AMIGA 3000

Multi-media voor een bodemprijs

Commodore is één van de grondleggers van de multi-mediagedachte. Het is dan ook niet verwonderlijk dat men de stap tot het uitbrengen van een audiovisuele alleskunner heeft genomen. De Amiga 3000 biedt gewoon het neusje van de zalm op hypermediagebied. Wel opmerkelijk is de scherpe prijs waar de Mac en de PC in dezelfde uitvoering vooralsnog niet aan kunnen tippen. Of Commodore met de Amiga 3000 werkelijk een flinke taartpunt van de multimedia-markt zal inpikken valt nog te bezien. Wel kan Commodore-Info u vast kennis laten maken met dit digitale hoogstandje. Al jarenlang wordt er door de professionele dtv-markt aan de Amiga getrokken. Met name de Amiga 2000 was erg in trek bij videoproducenten, kabelexploitanten en omroepen. Dan ging het echter vaak om een verbouwd 2000-model. Allerlei broadcast quality video-hardware moest tegen vrij forse bedragen in de 2000- systeemkast gezet worden. Met PAL-encoder, genlock, HIRES non- flickering videokaart, 68020/30-turbokaart, mathematische coprocessor, meerdere MB's aan vrij video-RAM en een zware harddisk zat men dan al snel rond de *f* 10.000,-.



Ruim een jaar geleden (in aankondigingen al meer dan twee jaar geleden) kwam Commodore met het Model 2500 de roep om meer up-todate Amiga's gedeeltelijk tegemoet. Het illustere chipdrietal Paula, Fat Agnus en Denise werd wat opgepept en de Amiga kreeg standaard een Motorola 68020 A2630 turbokaart en A2091 SCSI-controller met harddisk in de systeemkast. In een aantal modellen zat ook al een Flicker Fixer-videokaart. Menige gebruiker was daar al ruimschoots tevreden mee. Professionals bleven echter morren dat het hier om een verbouwde 2000 en niet om een gemoderniseerd Amiga-concept ging.
Het gemoderniseerde Amiga-concept is er nu wel. De 16-25 MHz Amiga 3000 mag een complete revisie van de Amiga-hardware genoemd worden. Een machine die voorbestemd is de multimedia en zakelijke presentatiemarkt in te palmen. Hoogwaardige systeemintegratie, verbetering van de videoprestaties en een scherpe prijsstelling moeten het in deze gaan maken. We geven in Commodore-Info een eerste impressie van deze jongste Amiga-telg.

Uiterlijk en systeemkast

De styling van de Amiga 2000-systeemkast is fraai gemoderniseerd. Het evenwicht tussen het wat bredere bovendeel van de kast en het wat smallere onderstel met ventilatiesleuven oogt nu veel beter. De enige zichtbare diskdrive-opening zit nu in het midden in een lange driehoekige uitsparing aan de voorzijde van de systeemkast. Uiterst links in deze sleuf vindt u tevens de indicatielichtjes voor de harddisk en power.
De muis en het toetsenbord zijn nauwelijks veranderd. Alleen zijn de aansluitingen voor joystick, muis en keyboard naar de rechterzijde van de Amiga-kast verhuist. Wel gewijzigd is de display in de vorm van de 14 inch Commodore 1950 VGA die de oude vertrouwde 1084 vervangt.
Aan de achterzijde van de systeemkast vinden we de connectors voor 23-pins standaard Amiga-RGB-video (15,75 KHz), 15-pins VGA-monitor (31,5 KHz), parallelle printerpoort, 25-pins SCSI- interface (extern), diskdrive en seriële printerpoort.
In de kast zitten twee 3.5 inch drives: een 880 KB diskdrive en een harddisk. Er is nog plaats voor een externe interne 3.5 inch drive.

De hoofdkaart

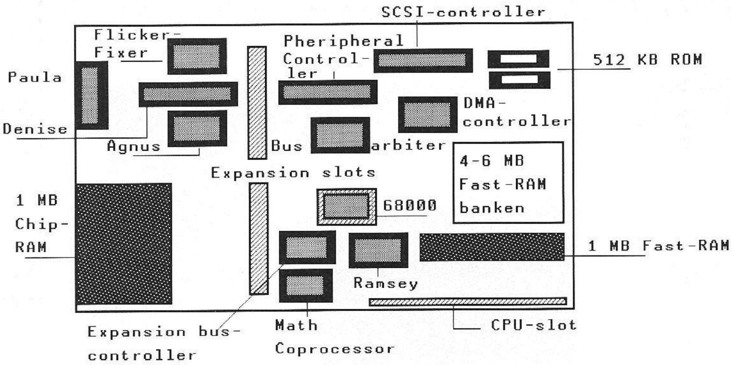
Wie bekend is met de oude Amiga-2000 hoofdkaart denkt op het eerste gezicht een geheel andere computer dan de Amiga 3000 geopend te hebben. Dat klopt ook want de oude motherboard-architectuur werd grotendeels verlaten en nu door een hoogwaardig geïntegreerde 32- bits open systeemarchitectuur vervangen. Alleen het grafische systeem gebruikt nog een 16-bits bus. De rest is, in tegenstelling tot de voorgaande Amiga-modellen allemaal 32-bits.

Behalve de hierna besproken drie vernieuwde custom chips bevat de hoofdkaart de volgende nieuwkomers:

- Een buscontroller
- Een met de SCSI-controller geïntegreerde 32-bits DMA- controller
- Een RAM-controller (koosnaam dubbed Ramsey) die het Chip-RAM (maximaal 2 MB) en 4-16 MB aan Fast-RAM bestuurt
- 512 KB in twee ROM-chips met het Amiga-DOS, system- libraries en multi-tasking routines
- Flicker-fixer-chip die een einde maakt aan de irritante beeld-flickering bij de hogere Amiga-resoluties; de hoogst bereikbare PAL-videoresolutie is 768 x 576 beeldpunten bij non- interlaced overscan

De CPU is zoals reeds werd opgemerkt de Motorola MC 68030 die afhankelijk van de meegeleverde mathematische coprocessor 68881 / 68882 op 16 / 25 MHz draait. Opmerkelijk is dat de van de Amiga 2000 en 2500 bekende Amiga- en PC-uitbreidingsslots op de hoofdkaart zelf ontbreken. Die zijn nu vervangen door twee zogenaamde backplane connectors waarin een zelfconfigurend uitbreidingsplateau gestoken wordt. In dit plateau gaan drie Zorro compatibele kaarten, twee AT-kaarten en een optionele video-adapter. De uitbreidingskaart staat verticaal in de beide 100-pins uitbreidings-sockets. Daardoor komen de connectors voor optionele Amiga Zorro III- en AT-kaarten horizontaal te staan. Uitbreidingskaarten worden nu dus horizontaal in de Amiga 3000 gezet in tegenstelling tot de verticale positionering bij de modellen 2000 en 2500. De naar keuze als 16- of 32-bits inzetbare uitbreidingskaart maakt aanpassing aan diverse hardware-systemen mogelijk en neemt dank zij de horizontale positie van de interfacekaarten minder kastruimte in. Wie **onder MS-DOS** wil draaien kan één van de Commodore (of van alternatieve leveranciers) Bridgeboards in een AT-slot steken. Dan kan de gebruiker zowel met Amiga- als MS-DOS aan de slag.

Het videoslot is geschikt voor de installatie van een aparte display-adapter, genlock of special effects generator. De voor de Amiga 2000 en 2500 gangbare genlocks passen hierin. In de toekomst verwachten we echter nog speciale 3000-versies voor specialistische videodoeleinden.



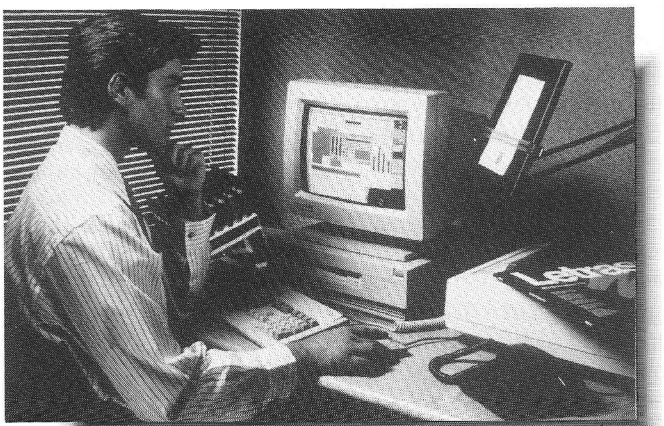
Verder werd al rekening gehouden met een toekomstige MC 68040- kaart. Een 200-pins connector staat reeds te wachten.

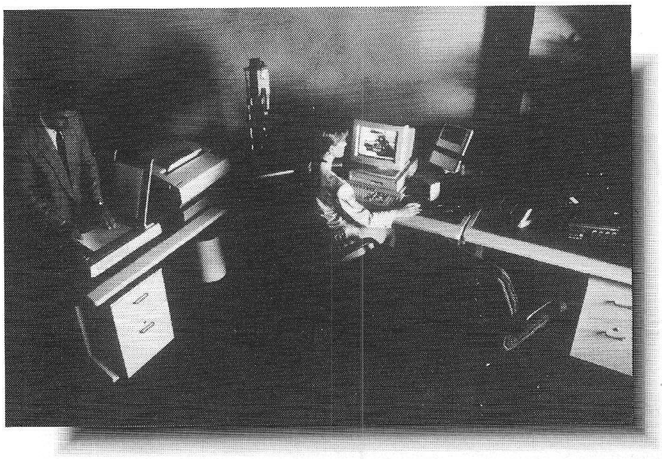
De custom chips

Het Chip-trio Denise, Agnus en Paula onderging weer de nodige verbeteringen. De **Super Fat Agnus** kan dubbel zoveel Chip-RAM, 2 MB, als de gewone Fat Agnus aan. Voor bezitters van de modellen 500 en 2000 is upgraden van Fat naar Super Fat helaas niet mogelijk.

(N.B.: **Chip-RAM** is dat gedeelte van het RAM dat zowel door de drie customchips als de CPU benut wordt. In dit RAM-deel worden met name de graphics en de geluidsfragmenten bewerkt. Uitbreiding is een wezenlijke voorwaarde voor HIRES-graphics, complexe animaties en geluiden. Het gedeelte dat alleen door de CPU gebruikt wordt noemt men **Fast-RAM**.)

De videochip **Denise** kan nu maximaal 1280 x 512 pixels in vier kleuren en bij 1280 x 265 pixels in 64 kleuren op de monitor zetten. Ook nieuw is de vier kleuren 640 x 480 non-interlaced modus. Verder beschikt de Amiga 3000 natuurlijk over het gehele palet van 4096 kleuren en alle overige videomodi vanaf 640 x 200 pixels en overscan voor videobeelden. Daar Denise volledig programmeerbaar is kan een softwareontwerper verschillende horizontale en verticale resoluties met elkaar mixen. Verder is het model 3000 voorzien van een **display enhancer** die een prima VGA-beeldje (31,5 KHz) op hiermee compatibele VGA- en multisync- monitoren zet. Zonde is dat van al deze schitterende beeldkracht bij de videotoepassingen niet zo veel meer overblijft. Het PAL (en NTSC) belemmeren gewoon door op de 15,75 KHz uit de Amiga RGB-uit-





gang. Commodore kan de gangbare videostandaard nu eenmaal niet verbeteren. De prestaties op VGA zijn daarentegen subliem.
Of de software het allemaal wel kan bijbenen blijft een tweede. De paletbeperkingen van het aantal kleuren blijven tot nieuwe releases uitkomen beperkt tot 32 in low en 16 in high resolution. Of u moet van de anti-aliased HAM of een van de halfbright-modussen gebruik maken. In de nabije toekomst zal voor het multi-mediagebeuren integratie van 8- en 24-bits kleuren nodig zijn.
Paula werd zover ons bekend alleen wat aan de beide andere herziene chips aangepast. Belangrijke vernieuwingen hebben zich bij deze customchip blijkbaar niet voorgedaan.
De nieuwe **Flicker-Fixer**-chip werd al besproken.

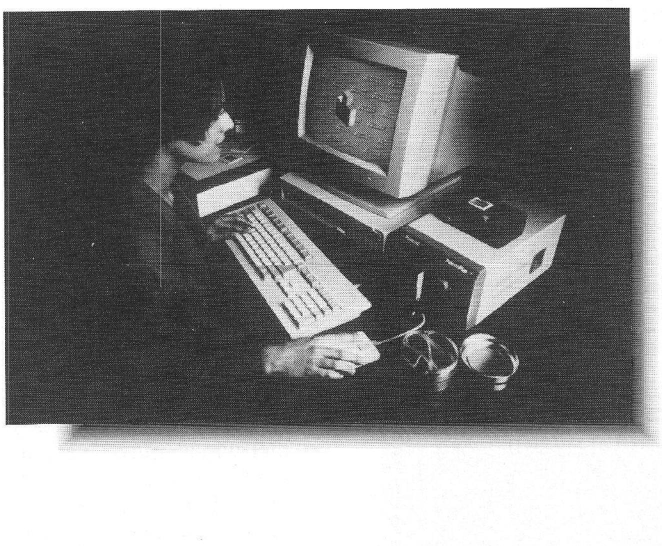
Harddisk en controller

Commodore biedt de keuze uit diverse **harde schijf-typen**. In Nederland gaat het om 3.5 inch modellen met een accestijd van 19 ms en een datacapaciteit van 40 of 100 MB. In de VS wordt ook een 50 MB type geleverd.
De **controllerchip** werkt met 32-bits DMA en het SCSI-protocol. Daarmee zijn imponerende harddisksnelheden mogelijk.

Software

De Amiga 3000 komt met **Amiga OS 2.0** en de **Workbench 2.0** in ROM. De veranderde werkbank springt het meest in het oog. Verbeterde schermgraphics en kleuren maken dit Amiga-gui een lust voor het oog. Multi-tasking is een fluitje van een cent. Er zijn betere Preference-tools, menu's, Adobe-fonts, bitmap fontscaling, verbeterde vensters, een krachtiger front-to-back utility en het is nu mogelijk om bestanden zonder gebruik van info-files te bekijken.

Het Amiga-DOS kan nu ook direct uit de Workbench gebruikt worden. Gewoon een datafile in de applicatie "droppen" en u kunt zonder DOS-shell aan de slag.
Standaard file- en fontrequesters behoren nu tot de basisuitrusting van het OS. Zelf programmeren van gadgets in het windows-systeem is aanmerkelijk eenvoudiger geworden. En natuurlijk is ook de interproces-communicatietaal **ARexx** present.
In het verleden was Commodore bepaald niet het toonbeeld van consequent compatibiliteitsgedrag. Bij de introductie van een nieuw OS en gui houdt je dan ook onwillekeurig je hart vast. Zal de oude software ook op de Amiga 3000 werken? Voor versie 1.3 geschreven programma's blijken, voor zo ver ons bekend, goed te werken. N.B.: Desgewenst kunt u de versies 1.3 van het besturingssysteem gewoon van de harddisk laden. Bij oudere versies kunnen storingen voorkomen. Koop daarom alleen de nieuwste releases.
Het UNIX-gebeuren staat nog niet echt in de kijkerd. Een Amiga 3000 zou prima als de spil in een klein UNIX-netwerk kunnen functioneren. Er begint langzaam iets te komen, maar echt grote toepassingen zijn er nog niet. Volgens Commodore zijn grotere UNIX-implementaties op korte termijn te verwachten.



Het multi-media auteurspakket **Amiga Vision** wordt elders uitgebreid besproken. Daarom hier slechts een vermelding.

Toepassingen

De scherpe prijsstelling, u koopt al een Amiga 3000 voor ongeveer *f* 10.000,- (= inclusief BTW), en de indrukwekkende prestaties maken deze multimedia-computer met name geschikt voor:

- Computer Ondersteund Onderwijs (COO); van eenvoudig thuis achter de PC studeren tot complete technische handboeken
- DeskTopVideo
- DeskTopPublishing
- CAD(D)/CAM en andere grafische ontwerpen
- UNIX-netwerken
- Imageprocessing
- Muziek in vier-kanaals-stereo en via de MIDI-mogelijkheden
- Wetenschappelijk onderzoek (simulaties, meet- & regeltechniek, beeldanalyse)

De Amiga is van alternatieve dtv-computer en hobbymachine omhoog gerezen tot een echt professionele presentatie-machine. In de wereld van multi-media, dtv, grafisch ontwerpen, vermaak, COO en muziek zal deze Amiga zijn weg zeker wel vinden. Of het model 3000 ook op andere toepassingsvlakken zal doorbreken valt echter nog te bezien.
Voor alle info: 020-882222.

U.S.

DELUXE VIDEO III

Compleet herschreven

Deluxe Video is al ruim 4 jaar een begrip in de dtv-wereld. De oerversie was een van de eerste video-softwarepakketten die erin slaagde titels, achtergronden, bewegende objecten en muziek tot complete miniprodukties aan elkaar te plakken. In tegenstelling tot de toen gangbare professionele grafische en titelapparatuur gebeurde dat nog zeer goedkoop en gebruiksvriendelijk ook. Voor rond de *f* 290,- werd de Amiga tot een complete grafische, titel- en animatiestudio uitgebouwd. Alle effecten, titels, muziekfragmenten, bewegende objecten, voor- en achtergronden werden keurig met de muis in een tijdbalk gezet. De Amiga nam het reken- en afspeelwerk geheel van de videomaker over.

Oude gebreken

Bood Deluxe Video bij de introductie veel, in de loop der jaren is het pakket toch flink belegen geworden. Aan moderne dtv- software worden hogere eisen gesteld dan de eerdere Electronic Arts-releases bieden. Veelvuldige punten van kritiek zijn:

- De incompatibiliteit met standaard .IFF-bestanden
- Het gebrek aan paletkleuren
- Incompatibiliteit met de moderne Amiga-videomodi
- Hakkelende en flikkerende animaties

Wat vroeger het summum van Amiga-kunnen was wordt nu door de verwende Amiga-dtv-er als ouderwets en irritant ervaren. Een reden voor Electronic Arts om het gehele pakket te herschrijven.

Wat is nieuw?

Bij **Deluxe Video III** zit alles voor de moderne dtv-studio erop en eraan. Het pakket lijkt dan ook helemaal niet



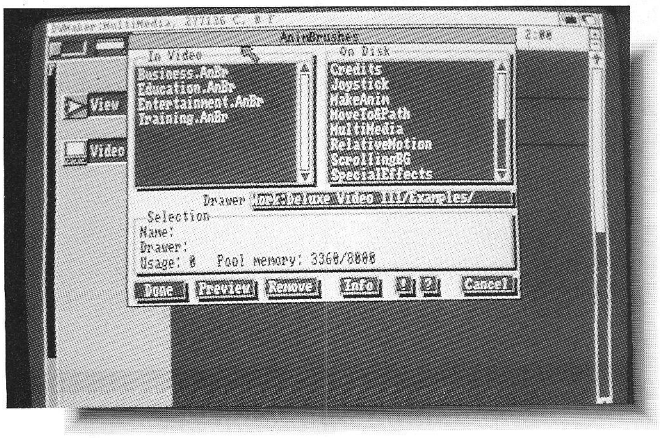
meer op de oude releases en is daar voor een groot deel incompatibel mee! De belangrijkste verbeteringen zijn:

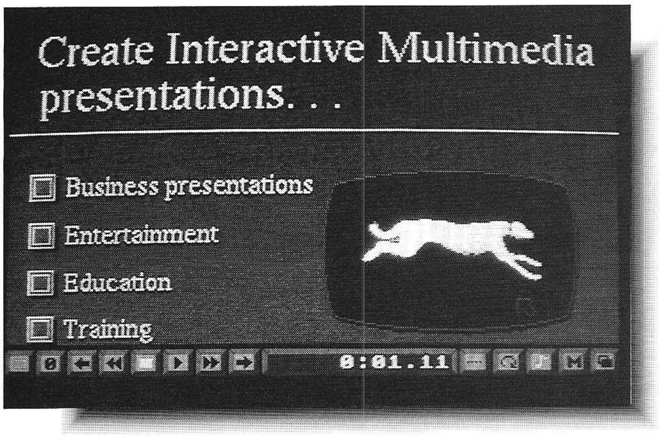
- Compatibel met **alle Amiga-videomodi**
- Hoeveelheid beschikbare kleuren, **maximaal 4096**, alleen nog maar hardwarematig beperkt
- **Double-Buffering**: de graphics van het volgende grafische beeld worden al tijdens de huidige display op de achtergrond in het video-RAM gezet; dat geeft een veel vloeiender animatie of beeldafwisseling
- De **dual playfields-aanpak** is geheel verlaten
- De **special-effects** zijn aanmerkelijk uitgebreid; er zijn nu meer dan 30 combinatiemogelijkheden met wipes, move, vergroten/verkleinen, roteren, fades in/out, mozaïeken, scrolls en maskers; wie nog meer wil zal naar een speciaal SEG-pakket zoals bijvoorbeeld Animagic moeten uitkijken
- Volledige **ARexx-integratie** voor besturing van genlock en videorecorder
- Compatibiliteit met **.IFF-ANIM-files** en **ANIM-brushes**. Deluxe Video III kan nu zelf ook gemaakte video's in ANIM-bestanden converteren
- **MIDI-besturing** van aangesloten electronische muziekinstrumenten biedt vele creatieve mogelijkheden voor de soundtrack

Kortom, Deluxe Video III is weer helemaal up-to-date.

Dtv op de Amiga

De studiofaciliteiten van Deluxe View III zijn grotendeels hetzelfde gebleven. Nog steeds start de gebruiker een **Video Script** en koppelt daar de benodigde **Scene Scripts** aan vast. De Scene Scripts zijn op hun beurt





weer opgebouwd uit **Effect Boxes** die voor- en achtergronden, objecten, titels, bewegingen en (special-) video-effects kunnen bevatten. Een Video Script bestaat dus uit diverse scenes die later in de gewenste volgorde worden afgespeeld. Die **volgorde bepaalt** de dtv-er aan de hand van de tijdsbalk of -lineaal waarop het begin en eind van de verschillende effecten gezet worden.
Het lijkt allemaal wel wat op een uitgebreid storyboard in een soort tijdlijst. Het gehele script is aan de basis opgebouwd uit kleine Effect Boxes die volgens het tijdschema worden opgeroepen. Gezamenlijk leveren deze Effect Boxes één videoproduktie die later geheel automatisch afgespeeld kan worden.
Voor het animeren van objecten bestaan ruwweg twee mogelijkheden:

- Op de tijdsbalk de verschillende bewegingen aan elkaar koppelen
- Bij niet-lineaire bewegingen deze met de muis voortrekken of direct als coördinaten invoeren

Op het eerste gezicht lijkt een complexe script-structuur zoals bij Deluxe Video III RAM te vreten. Gelukkig valt dat in de praktijk reuze mee. Dank zij een slim RAM-algoritme wordt dubbelladen van gezamenlijk bruikbare voor/achtergronden, objecten, letters en muziekfragmenten voorkomen. Bovendien worden niet meer noodzakelijk geachte stukken video direct uit het geheugen verwijderd. Op deze wijze kunnen redelijke grote script-files vloeiend in een videogeheugen van 1 MB lopen.

Toepassingen

Deluxe Video III richt zich in de eerste plaats op de semiprofessionele dtv-markt. Een ieder die voor weinig geld zijn of haar videoprodukties eens smaakvol met titels wil aankleden of met computergraphics wil verluchten is bij dit Electronic Arts-videopakket aan het juiste adres. Andere toepassingen kunnen zijn: slideshows, reclameprodukties, simulaties en interactieve onderwijsvideo's.
De tekenfilmmaker kan tezamen met een goed paintpakket prima aninimatiefilms creëren. Verder kan digitizersoftware goed van pas komen voor het maken van spectaculaire effecten.

Praktijk

Het leren van Deluxe Video III kost onvermijdelijk tijd.



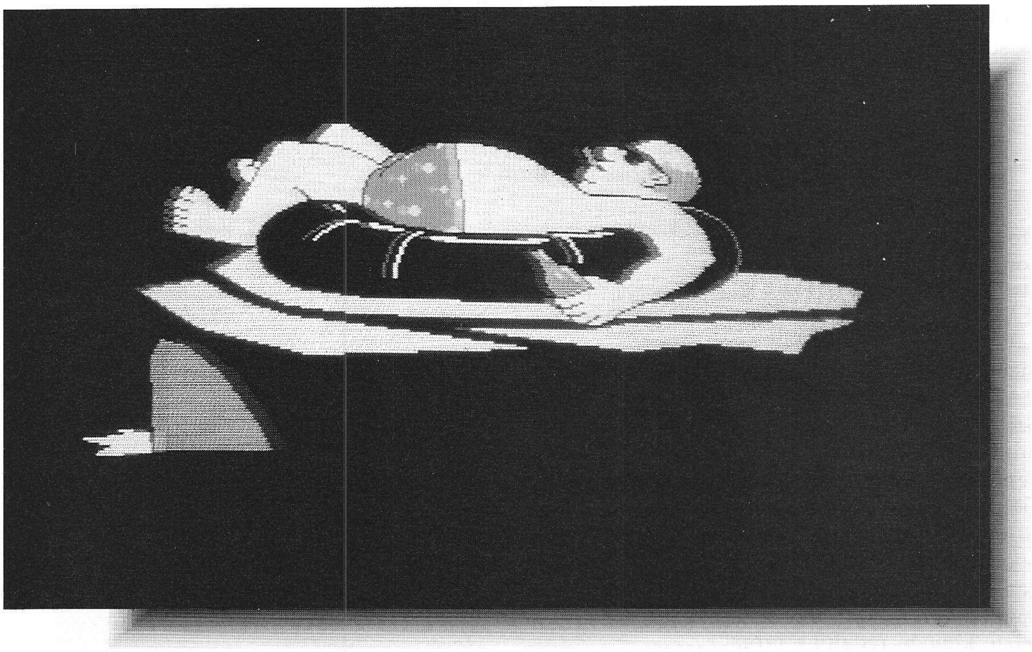
Echter wel goed gespendeerde tijd. Het duidelijke handboek met verhelderende leervoorbeelden helpt zelfs de dtv-leek binnen één dag op weg naar imponerende dtv-prestaties.
Wat u uiteindelijk allemaal met Deluxe Video III kunt doen hangt grotendeels van de eigen creativiteit en de beschikbare hardware af. Vast staat dat voor bijna *f* 300,- een complete dtv-studio in huis komt die in geval van professionele hardware al gauw 50 maal zo veel kost. Aanbevolen voor de home-dtv-er en maker van interactieve produkties.

U.S.



FANTAVISION

Vergeten eenvoud?



Het video-animatiepakket Fantavision van Broderbund is al wat langer op de markt. Je hoort er echter slechts weinig over en dat is jammer, want de software biedt veel gebruiksvriendelijke DTV-mogelijkheden voor relatief weinig geld. Natuurlijk is Fantavision qua mogelijkheden geen Deluxe Video III, Sculpt 3D of Deluxe Paint III. Deze (semi-)professionele pakketten zijn niet alleen dubbel zo duur, maar zeker ook dubbel zo lastig om onder de knie te krijgen. De beginner die eens een leuk animatiefilmpje wil maken, de home-videoprodukties gaat verbeteren of een (wervende) attractieve display wil creëren, kan met Fantavision alle kanten uit.

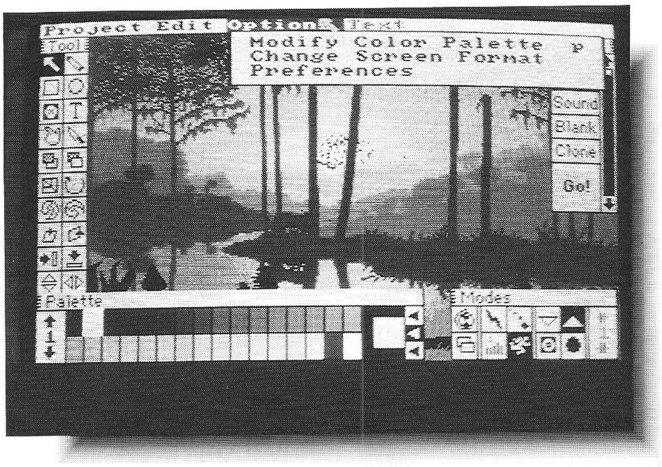
De Enhanced Fantavision-versie voor de Commodore Amiga valt in principe onder de noemer 2/3D tekenfilm-trucage-programma. In deze biedt Fantavision alles wat de Amiga-amateur zich maar kan wensen. Flitsende animaties (tot 128 verschillende tussenbeelden), zuinig geheugengebruik, attractieve kleurwisselingen, het gebruik van achtergronden, bewegende objecten plus teksten en bovenal een relatief eenvoudige maar doeltreffende methode van Amiga-video maken.
Opvallend zijn de vloeiend overlopende beelden. Zeker in deze prijsklasse. PAL, Overscan, Halfbright (niet Extra) en HAM staan als videomodus ter beschikking. Alle met andere teken- of digitizer-software gemaakte IFF-bestanden kunnen voor gebruik in Fantavision geladen worden.

Light, Camera, Action

Sinds de kinderschoenen van de filmindustrie is de frase licht, camera en actie een legendarisch concept. Wie hem precies bedacht heeft is mij onbekend. Ook nu gaat Light, Camera en Action zelfs bij een modern Amiga-pakket als Fantavision nog altijd op.
De Amiga 500 t/m 3000 functioneert bij Fantavision zowel als opname-camera, scene-ontwerper, lichtbron en actiemaker (animator) als afspeelprojector. Hard- en software vormen daarbij een, in deze prijsklasse, onovertroffen desktopvideoteam. Bij een IBM-compatibel PC, Apple Mac of professionele video-effectapparatuur bent u een veelvoud aan geld kwijt.

Werken met Fantavision

Na het opstarten van Fantavision (voor de 500- en 1000-gebruikers past alles geluk-



kig op slechts één diskette) verschijnt het openingsscherm. Bovenaan vindt u de vier hoofdmenu's: Project, Edit, Options en Text. Links staat de ikonenbalk met teken- en bewerkings-tools. Onderaan staan het kleurenpalet en de verschillende modes. Rechtsboven staan in het filmvenster de filmtracks met afspeeloptie.

Een basisanimatie valt heel eenvoudig te maken. Neem een object, zelf tekenen of uit een bibliotheek inladen, en voorzie dit van het juiste kleurenpalet. Zet dit object met de muis ergens op het scherm. Dan is frame 1 (het eerste beeldje) al gereed. Nu wordt het tijd om een tweede frame te maken. Wijs met de muis de optie Clone (juist ja, kloon) en het framenummer bovenin de beeldteller schiet nu op 2. Kies de pijl uit de toolbox, er verschijnt een "grijphandje", pak het object en beweeg dit schermobject naar de gewenste lokatie. Bij het afspelen beweegt het object nu vloeiend naar deze lokatie en weer terug. Eenvoudiger kan het niet.

Eenvoudige bewegingen vervelen echter al snel. De verwende DTV-er en zijn of haar publiek wil gewoon meer. Zoiets kan bijvoorbeeld via een transformatie, het ene object in het andere laten overgaan. Een simpele methode daarvoor is het gebruik van de tekentools bij de opeenvolgende frames. Als voorbeeld een 3-frame-movie. Teken eerst een object, kies vorm en kleur. Teken een tweede object van een andere vorm en kleur. Frame 1 is nu klaar.

Klik met de muis op de Blank-optie van het filmvenster. Teken nu weer twee objecten, bijvoorbeeld een rechthoek en een cirkel, met behulp van de tekenopties. Frame 2 is nu klaar.

Frame 3 wordt weer gemaakt door op Blank te klikken en een of meerdere figuren te tekenen. Bij het afspelen zorgt fantavision er geheel automatisch en vloeiend voor dat de verschillende geometrische figuren in elkaar over gaan en van kleur veranderen.

Uiteraard wordt het allemaal wat moeilijker bij het gebruik van gecompliceerdere animatieobjecten en meerdere frames. Door experimenteren wijst de weg zich gelukkig snel vanzelf.

Het Frame Window

Centraal in Fantavision staat de regiestoel van het Frame Window. In dit venster staan de volgende opties en indicaties:

- bovenaan de frameteller (beeldteller);
- daaronder de Movie-info met alle gegevens over objecten en tracks;
- de soundtrack;
- Blank voor het invoegen van een nieuw frame;
- Clone voor het klonen en aan het animatie-einde plaatsen van een frame;
- GO! voor de playback;
- en rechts de scrollbar voor het instellen van de snelheid.

Met deze animatie-tools kunt u behoorlijk gecompliceerde tekenfilms in elkaar sleutelen. Maximaal passen 16 objecten van elk 24 punten per object. Deze objecten worden getypeerd door hun door Fantavision toegekende objectnummer en het aantal points. De software houdt alle aan de objecten toegekende bewegingen en kleurveranderingen nauwkeurig bij. Als gebruiker heb je er geen omkijken meer naar.

Voor het los, zonder Fantavision in actie, playbacken kan dit Broderbund-pakket een zelfstandige showdisk aanmaken. Daartoe behoeft de gebruiker slechts de Fantaplayer-ikoon en de gewenste movies naar de target-disk te kopiëren.

De tools

Voor het tekenen van objecten beschikt Fantavision over een reeks tekengereedschappen. Deze tools omvatten de tekenpen (potlood), de editor-pointer, grab-point-tool, geometrische figuren (cirkel, rechthoek), bitmap (voor het maken van bitmap- files van het beeldscherm, text, insert point (voor het veranderen van vormen), delete point, zoom send to front/to back, turn, rotate, lean (hellen over horizontale of verticale as), squash (platdrukken en uitrekken), flip, cut, paste en copy. Kortom een compleet tekenprogramma uit de eenvoudiger klasse.

Bij het maken van simpele animatie-objecten bieden de eigen Fantavision-tools voldoende soelaas. Voor ingewikkelder objecten komen specialistische IFF-tekenpakketten in aanmerking. Hetzelfde geldt voor een geschikte achtergrond die via het Project-menu kan worden binnengehaald.

De soundtrack

Net als de andere Amiga DTV-software is ook Fantavision voorzien van een soundtrack. De geluiden worden via de Sound Setup Requester aan de afzonderlijke frames gekoppeld.

De geluiden zelf haalt u uit de Fantavisions-sound-library of externe bron (andere software, samplers). Tot de instelmogelijkheden van de requester behoren: Channel 1/2, Volume, Balance, Echo, Pitch, Duration, Playback-Speed en Sustain. De overige opties zijn zoals gebruikelijk Add, Save, Remove, Play, Cancel en OK.

Conclusie

Fantavision biedt voor rond de *f* 150,- meer dan menig freeware DTV-pakket. Goed, het is geen Deluxe Video of Aegis DTV-pakket, maar de eenvoud voor het creëren van simpele animaties maakt veel goed. Bij het scheppen van complexe tekenfilms wordt ook Fantavision bewerkelijk en moeilijk. Aanbevolen voor eenvoudige dtv, promoties, kabelkranten en interactive video-educatie. Gezien de prijs, de mogelijkheden en het gebruiksgemak zou dit Amiga DTV-pakket best eens wat meer aandacht mogen krijgen.

Verkrijgbaar bij onder andere Salasan.

U.S.

ALLADYNE VGS

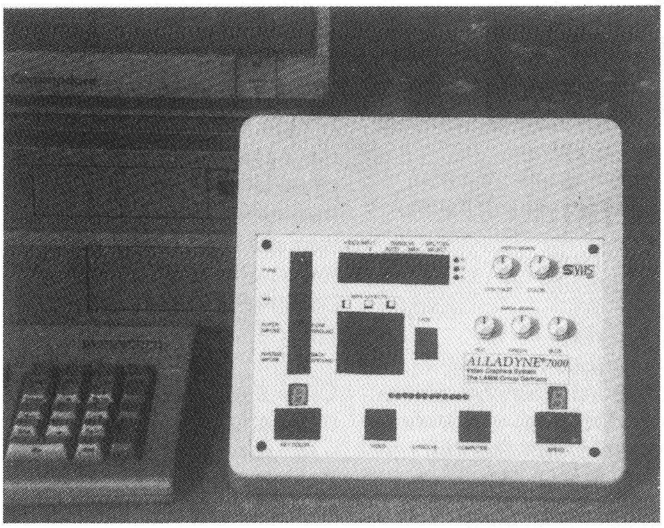
Volwaardige Amiga-videostudio

Er verschijnen steeds meer geïntegreerde videosystemen voor de Amiga-lijn. Ook het door Peter Biet aangeboden Alladyne Video Graphics System (VGS) is een combinatie van een genlock, videodigitizer, videoprocessor en RGB-splitter. Het apparaat is modulair opgezet, zodat later altijd uitbreiding mogelijk blijft.

De video-amateur die het Alladyne VGS aanschaft krijgt een compleet videoconcept in huis. Analoge videotechniek en digitale RGB-processing ontmoeten elkaar in één compact kastje.

Vier functies

De genlock is gebaseerd op het zo succesvolle Sony-chip-concept dat ook bij de Commodore 2301 genlock toegepast is. Deze kaart-genlock geeft een schitterende beeldkwaliteit, maar helaas ook niet meer dan dat. Besturing van de genlock-functies is niet mogelijk. Daar brengt Alladyne nu verandering in. Vanuit het regelpaneel kunt u alle RGB-, inverse/impose-, superimpose- en achtergrondkleur-functies besturen. De genlock biedt naar keuze S-VHS (Y/C)-output of standaard FBAS-composietvideo. Natuurlijk biedt de Y/C-uitgang de beste beeldkwaliteit.
De ingebouwde videoprocessor biedt in totaal zes (drie vaste met twee combinaties) wipe-effecten. Het RGB-splitterdeel zorgt voor een afzonderlijke digitalisering in de kleuren rood, groen en



blauw bij het gebruik van een videodigitizer.
Als videodigitizer kunt u kiezen uit Digi-View 4.0 of De-luxe- View.
Voor de semi-professionele gebruiker is een videoprocessor voor de correctie van het oorspronkelijke videosignaal leverbaar. Daarmee zijn onder andere contrast, helderheid, kleurverzadiging en fade in/out regelbaar.

Effecten

Wat kunt u nu eigenlijk allemaal met het Alladyne VGS doen? Dat hangt deels af van het gekozen type en de rest van de eigen creativiteit. De basis-unit Alladyne 3000 biedt de genlock (5.5 MHz bandbreedte, signaal-ruisafstand beter dan 73 dB) met een RGB-splitter (LED-indicatie per kleur) en een kleine special-effect generator. Daarmee zijn mogelijk:

- genlocking: impose, superimpose, inverse-impose, videomix en videodissolve (overvloeien);
- manuele RGB-splitting voor een videodigitizer;
- met behulp van een titelpakket professionele video- aftiteling;
- wipe-effecten (zes in totaal).

In de basisuitvoering kost het Alladyne VGS circa *f*1.200,-. De duurdere Alladyne 5000, circa *f*1.900,-, beschikt over een ingebouwde videoprocessor. Daarmee kan de videomaker het oorspronkelijke videosignaal bewerken. Mogelijk zijn onder meer:

- het bijregelen van de helderheid (lichter of donkerder);
- contrastbeïnvloeding;
- beïnvloeding van kleur en kleurverzadiging;
- diverse fades: ook in combinatie met mix, superimpose en inverse-impose; de impose-kleur is naar vrije keuze.

De ingebouwde RGB-splitter wordt op de muispoort 2 aangesloten en werkt automatisch op stilstaande videobeelden. Desgewenst kunt u dit ook via de met LEDs gemarkeerde tiptoetsen.
Het topmodel, de Alladyne 7000, biedt werkelijk 40 videorovers. Nu beschikt de videomaker tevens over een ingebouwde Digi-View- Gold of Deluxe-View-digitizer plus het befaamde titelprogramma Video-Page. Kortom een complete videostudio naast de Amiga. De prijs van het 7000-model ligt rond de *f*2.500,-.

Praktijk

In de videopraktijk geeft het Alladyne-systeem alleszins bevredigende tot goede resultaten. De beste resultaten krijgt men uiteraard bij Super-VHS. Ruim voldoende voor semi-professionele en low-budgetprodukties. Menige amateur en reportagemaker zal niet meer nodig hebben. Wie echt meer wil zal naar aanmerkelijk duurdere randapparatuur moeten uitkijken.
Voor verdere informatie; Peter Biet Computerdesign. Dietershausener-Strasse 28, 6409 Friesenhausen, Duitsland.

U.S

AMIGA C

In de vorige twee afleveringen zijn de windows en screens behandeld. Nu u in staat bent om zowel een window als een screen te openen, begint het interessant te worden om iets met deze informatie te doen. Een window of screen zou nutteloos zijn als in dit window en screen geen data afgebeeld zouden kunnen worden. Het doel van deze aflevering zal nu wel duidelijk geworden zijn. We gaan op Intuition-niveau beschrijven hoe binnen windows tekst en grafische data afgebeeld kunnen worden.

Intuition support

Intuition ondersteunt tekst en grafische afbeeldingen door middel van een aantal functies. Om deze functies te kunnen gebruiken dient er een vrij grote hoeveelheid informatie opgegeven te worden, zoals de kleur van een afbeelding, de x-en y-positie, etcetera. Als deze informatie in de parameterbody van een functie opgegeven zou moeten worden dan zou dit in een schier eindeloze parameterlijst resulteren. Gekozen is daarom voor een andere oplossing.

Images

In het include-bestand 'intuition/intuition.h' vindt u de definitie van de Image-structure, welke ook terug te vinden is in afbeelding 1. In de Image-structure moeten alle specifieke kenmerken van het Image worden vermeld. Is dit gebeurd dan kan het Image in een Window worden afgebeeld, echter voordat we dit gaan doen geven we eerst een beschrijving van de verschillende elementen van de Image-structuur.

Afbeelding 1. De Image-structure

```
struct Image {
  SHORT LeftEdge;
  SHORT TopEdge;
  SHORT Width;
  SHORT Height;
  SHORT Depth;
  USHORT *ImageData;
  UBYTE PlanePick;
  UBYTE PlaneOnOff;
  struct Image *NextImage;
};
```

De eerste twee parameters van de Image-structuur, LeftEdge en TopEdge bevatten de coördinaten van de linkerbovenhoek van het Image. Deze coördinaten zijn relatief ten opzichte van het plaatsingsvenster waarbinnen het Image moet worden afgebeeld (zie afbeelding 2). Hierop komen we later terug bij de beschrijving van de functie DrawImage.

Afbeelding 2. Relatieve positie van Image tov. van venster.

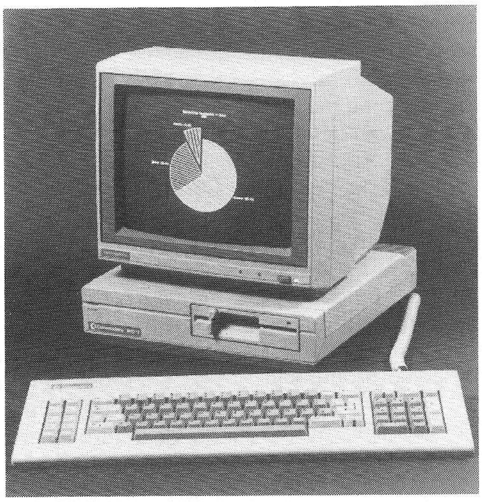
x,y positie opgegeven bij DrawImage()

x,y positie van Image, relatief tov. DrawImage()-venster

Image

16 bits-grenzen

Met Width wordt de breedte van het Image ingesteld, Height geeft de hoogte van



het Image aan. Hierbij dient opgemerkt te worden dat hier een ruwe schatting van de breedte en hoogte van het object moet worden opgegeven. U dient de waarden af te ronden op 16 bits grenzen. Om dit te illustreren volgt hier een klein voorbeeld.

Stel, u heeft een Image welke 14 bij 29 bits groot is. Dit dient u af te ronden naar 16 bij 32. Deze waarden worden dan vervolgens in de Image-structuur ingevuld.

Depth dient de diepte van het Image te bevatten. Om uit de diepte het aantal kleuren af te leiden, herhalen we de formule hiervoor nog eens,

2 ^ diepte = aantal_kleuren

Het element ImageData bevat, zoals de naam al doet vermoeden, een pointer naar de Image-gegevens. Deze data moeten in het Chip-RAM geplaatst zijn.

Hoe definiëren we nu een Image? Bij de beschrijving van de elementen Width en Height is al naar voren gekomen dat de breedte en hoogte op 16 bits grenzen worden afgerond. Dit heeft te maken met het feit dat de Image's opgeslagen worden in UWORDS. Kijk nu eerst even naar afbeelding 3. Hier ziet u een vreemd aandoende afbeelding waarin 16 kleine vlakjes met en zonder X-en te onderscheiden vallen Dit moet de eerste regel van een Image voorstellen. De 16 vlakjes stellen 16 bits voor. De X-en zijn de bits welke gezet en dus zichtbaar zijn. Elke X is een 1 en kan naar een hexadecimale waarde worden omgerekend door per nibble, per 4 bits dus, de waarde te berekenen. Nu ziet u boven de vlakjes per nibble 0 tot 3 afgebeeld. Deze waarden stellen de bitnummers voor. Kijk nu naar nibble 'a'. Hierin zijn de eerste twee bits gezet. Om hiervoor de hexadecimale waarde te berekenen, bepaalt u de som van de gezette bits. Dus voor nibble 'a' wordt dit 2^0 + 2^1 = $03. Doe dit voor alle nibbles en u heeft alle data voor de Images berekend.

Afbeelding 3. Bepaling van Image gegevens

```
 a       b       c       d
3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0
Data
 abcd
XX XX  X X X
```

=$03 =$06 =$02 =$0a
=0x362a

PlanePick bepaalt welke bitplanes gebruikt worden. Aangezien de bitplanes bepalen wat de uiteindelijke kleur van het Image wordt, kan met PlanePick de kleur van het Image beïnvloedt worden.
Elk bit stelt een bitniveau voor. Wilt u alleen bitniveau 1 gebruiken dan zet u bit 1 op 1, terwijl de rest van het byte op 0 gezet wordt. Voor bitniveau 2 wordt bit 2 op 1 gezet, enzovoort.
PlaneOnOff. Met dit element kunt u niet-gedefinieerde bitplanes toch een functie geven. Elk aangeduid bitniveau wordt opgevuld met enen. Zo kunt u een achtergrondkleur genereren.
Als laatste element treffen we NextImage aan. Heeft u meer Images welke u in één keer afgebeeld wilt zien, dan moet in dit element de pointer naar de volgende Image-structuur staan. Een klein voorbeeld.

```
struct Image DemoImage1 = {
  0,0,32,16,&DemoData,
  0x01,0x00,NULL };
struct Image DemoImage2 = {
  48,0,32,16,&DemoData2,
  0x01,0x00,&DemoImage1 };
```

Hiermee onstaat de krachtige voorziening om in één keer een serie van afbeeldingen, lees Images, op het scherm af te beelden.

DrawImage

Is eenmaal bekend wat de eigenschappen van het Image zijn, dus zijn de elementen van de Image-structuur ingevuld, dan kan het Image op het beeldscherm worden gezet. Hiervoor kent Intuition de functie DrawImage(). Hoe ziet deze functie eruit?
DrawImage(pointer naar RastPort, Image structuur, x-positie tov. window, y-positie tov. window)
De pointer naar de RastPort valt eenvoudig uit de Screen of Windowstructuur te halen. Voor het actuele scherm of venster kan het er dan als volgt uit zien:

```
struct RastPort *S_RastPort;
struct RastPort *W_RastPort;

S_RastPort=&IntuitionBase->ActiveScreen->RastPort;
W_RastPort=&IntuitionBase->ActiveWindow->RPort;
```

Een aanroep van de functie DrawImage() zou dan, uitgaande van het voorgaande voorbeeld, als volgt kunnen luiden:

```
DrawImage(W_RastPort,DemoImage,100L,100L);
```

Na gebruik van dit voorbeeld zal het Image 'DemoImage' vanaf positie 100,100 binnen het actuele window tevoorschijn komen.

Vrij complexe lijnpatronen

Vanuit Intuition kan men op vrij eenvoudige wijze vrij complexe lijnpatronen, Borders genaamd, tekenen. Wat is een Border? Het kan zijn:

- een lijn,
- een willekeurige verzameling van lijnen,
- een rechthoek of vierkant.

Wil men een Border gebruiken dan dient men ook een structuur in te vullen. De Border-structuur, welke te vinden is in afbeelding 4, kent een aantal gelijkenissen met de Image-structuur. De eerste twee elementen namelijk, LeftEdge en TopEdge vervullen hier dezelfde functie als bij Images. Ook hier gaat het weer om een relatieve positie ten opzichte van een van tevoren bepaalde positie. Deze positie wordt ingesteld met behulp van de functie DrawBorder().

Afbeelding 4. De Border-structuur

```
struct Border {
  SHORT LeftEdge;
  SHORT TopEdge;
  UBYTE FrontPen;
  UBYTE BackPen;
  UBYTE DrawMode;
  BYTE Count;
  SHORT *XY;
  struct Border *NextBorder;
};
```

FrontPen is een wijzer naar een kleurenregister. Hierin wordt dus feitelijk de kleur van de lijn bepaald. De maximale waarde is afhankelijk van de diepte, en dus van het aantal beschikbare kleuren, van het scherm waarin men werkt. BackPen wordt op dit moment nog niet gebruikt voor Borders, dus kan altijd op 0 gesteld worden.
Met DrawMode wordt ingesteld hoe het lijnenpatroon op het scherm wordt getekend. Er staat een aantal modi ter beschikking waarvan de belangrijkste zijn:

- JAM1 : de lijn wordt op het scherm getekend met gebruikmaking van de kleur in FrontPen.
- INVERSEVID: de lijn wordt in z'n inverse component getekend.

Count geeft aan hoeveel getallenparen verwacht mogen worden.
Het element XY dient een pointer naar de punten-array van de Border-structuur te bevatten. Deze array dient paren van x- en y-getallen te bevatten. Deze x,y posities zijn relatief ten opzichte van de x,y-positie aangegeven in de Border-structuur.
Ook de Border-structuur kent de mogelijkheid om zichzelf aan een volgende Border te 'hangen'. Het element NextBorder dient dan een pointer naar een volgende Border-structuur te bevatten.

DrawBorder

Met één aanroep van de functie DrawBorder() kan men vervolgens een collectie van Borders op het scherm zetten. Het formaat van de functie DrawBorder() luidt als volgt:

DrawBorder(pointer naar RastPort, Border-structuur, x-positie t.o.v. window, y-positie tov. window);

Ook hier is weer een pointer naar de RastPort van het scherm of window nodig. Deze kan op dezelfde wijze worden verkregen als bij de Images. De overige drie parameters verduidelijken we met een voorbeeld.

```
SHORT DemoParen = {
  0,0,150,0,150,150,0,150,0,0
};
struct Border DemoBorder =
{
  0,0,1,0,JAM1,5,
  &DemoParen,NULL };

void
TekenBorder(WindowPtr)
struct Window *WindowPtr;
{
 struct RastPort
*DemoRPort;

 DemoRPort=&Window
Ptr->RPort;

 DrawBorder(DemoRPort,
DemoBorder, 50L, 20L);
}
```

De tweede parameter dient dus een verwijzing naar de te tekenen Border-structuur te bevatten. De derde en vierde parameter moeten ondertussen duidelijk zijn. Ze zijn al naar voren gekomen bij de Images.
Het mag duidelijk zijn dat de Borders zich niet lenen voor het afprinten van tekst. Het definiëren van één teken alleen al zou een heidens werk zijn.

Tekst in Intuition

Intuition kent voor tekst een andere functie en bijbehorende structuur. Ook deze structuur, IntuiText genaamd, vertoont vele gelijkenissen met de vorige twee behandelde structuren. De eerste twee elementen, zie ook afbeelding 5, FrontPen en BackPen bevatten respectievelijk de voor- en achtergrondkleur van de tekst. DrawMode geeft weer hoe de tekst afgebeeld moet worden. Ook hier is weer een aantal keuzemogelijkheden, waarvan de belangrijkste opgesomd zijn bij de Borders.

Afbeelding 5. De IntuiText-structuur

```
struct IntuiText {
  UBYTE FrontPen;
  UBYTE BackPen;
  UBYTE DrawMode;
  SHORT LeftEdge;
  SHORT TopEdge;
  struct TextAttr *ITextFont;
  UBYTE *IText;
  struct IntuiText *NextText;
};
```

De elementen LeftEdge en TopEdge bevatten de relatieve coördinaten ten opzichte van de basiscoördinaten. Deze worden ingesteld met de functie PrintIText().
Wanneer de tekst in een ander font moet worden afgebeeld, dan kan het element ITextFont uitkomst bieden. Het dient een pointer te bevatten naar een TextAttr-structuur. Hierop komen we in een latere aflevering terug. Voorlopig accepteren we het standaard font en vullen we hier NULL in.
Wil men uiteindelijk een tekst afprinten op het scherm dan zal IText een pointer naar een tekststring moeten bevatten. Men kan ook direct de tekststring opgeven zoals in het volgende voorbeeld geïllustreerd is:

```
DemoText.IText=(UBYTE
*)"Dit is een voorbeeldtekst";
```

De compiler zal hier het adres van de tekststring "Dit is een voorbeeldtekst" invullen zodat u dit niet zelf hoeft te doen.
Ook bij IntuiText bestaat de mogelijkheid structuren te koppelen. NextText dient bij meerdere IntuiText-structuren een pointer naar de volgende IntuiText-structuur te bevatten.
Het eigenlijke afbeelden van de tekst kan geschieden als de IntuiText volledig genitialiseerd is. Een aanroep van de functie PrintIText() is voldoende om een tekst op het beeldscherm tevoorschijn te toveren. Hoe ziet PrintIText() eruit?

PrintIText(pointer naar RastPort, IntuiTextstructuur, x-positie tov. Window, y-positie tov. Window).

Het formaat is gelijk aan dat van de vorige twee besproken functies. Daarom volstaan we hier dan ook met een voorbeeld om de parameters uit te leggen.

```
struct IntuiText DemoText = {
  1,0,JAM1,0,0,NULL,
  (UBYTE *)"Dit is een
demotekst",
  NULL };

void PlaatsTekst(WindowPtr)
struct Window *WindowPtr;
{
 struct RastPort
*DemoRPort;

 DemoRPort=&
WindowPtr->RPort;

 PrintIText
(DemoRPort,DemoText,50L,50
L);
}
```

Initialisatie van structuren

In de voorgaande voorbeelden werden de structuren geïnitialiseerd zonder dat de elementen werkelijk bij naam genoemd werden. We maakten gebruik van een compiler-voorziening die 'auto-initialisation' genoemd wordt. Deze voorziening kan behoorlijk handig zijn, doch heeft ook een fors nadeel. Alle variabelen worden immers globaal geïnitialiseerd. Dit is in strijd met één van de principes van het netjes programmeren. Men moet bij het programmeren namelijk streven naar het gebruik van zo weinig mogelijk globale variabelen. Dat dit bij de Amiga soms zeer moeilijk is, blijkt uit het feit dat voor de meest simpele zaken flinke structuren ingevuld moeten worden. Hiervoor bestaat niet één-twee-drie een oplossing, dus u staat voor de mogelijkheid om of het geheel van structuren in een include-file te 'auto-initialiseren' of u haalt alle definities op het lokale vlak en vult dus structuren op 'field-calling' basis, dus per elementnaam, in.
Om het verschil te verduidelijken geven we twee keer de initialisatie van een IntuiText structuur. De eerste keer met auto-initialisatie, de tweede keer op 'field-calling' basis.

- 1. auto-initialisation:

```
struct IntuiText DemoText = {
  1,0,JAM1,4,8,NULL,
  (UBYTE *)"Commodore
INFO Amiga C",
  NULL };

void
SchrijfTekst(WindowPtr)
struct Window *WindowPtr;
{

PrintIText(&WindowPtr->RPo
rt, DemoText, 100L,100L);
}
```

- 2. field-calling:

```
void
SchrijfTekst(WindowPtr)
struct Window *WindowPtr;
{
 struct IntuiText DemoText;

 DemoText.FrontPen=1;
 DemoText.BackPen= 0;
```

```
DemoText.DrawMode
=JAM1;
DemoText.LeftEdge=4;
DemoText.TopEdge=8;
DemoText.ITextFont=NULL;
DemoText.IText=(UBYTE
*)"Commodore INFO Amiga
C";
DemoText.NextText=NULL;

PrintIText
(&WindowPtr->RPort,DemoT
ext, 100L,100L);
}
```

Voor welke methode gekozen moet worden is een keuze van persoonlijke voorkeur, doch wilt u binnen het net-programmeren-principe blijven dan dient u optie 2 te prefereren boven optie 1. Het grootste nadeel van optie 2 is dat deze veel meer bewerkelijk is dan optie 1.

Het voorbeeld

Om nu het besprokene nog eens te illustreren hebben we een klein voorbeeldprogramma geprogrammeerd. Dit programma maakt veelvuldig gebruik van 'auto-initialisation', dus is niet bepaald netjes maar wel overzichtelijk.
Het programma loopt zowel met de Aztek 3.6 als met de Lattice 5.04 compiler.
Enige opmerkingen bij het programma. Het programma gebruikt de graphics.library. Deze library hebben we tot nu toe nog niet behandeld. Schenk er daarom niet te veel aandacht aan. De library is alleen nodig voor de twee functies LoadRGB4() en RemakeDisplay(). Deze functies zorgen ervoor dat de schermkleur aangepast wordt.
Compile- en link-instructies,
Aztek: cc cdemo.c +L
ln cdemo.o -lc32
Lattice: lc -L cdemo.c

Johan & Johan

Literatuurlijst

° Berry, J.T (1986) Inside the Amiga with C, Howard W.Sams & Comp., Indiana (USA) ISBN 0-672-22468-2

° Bleek, Jennrich & Schultz (1989), Amiga Intern 2.0, Bruna uitg., Utrecht, ISBN 90-229-3683-X

° Jennrich, Trapp & Weltner (1987), Het supergrafiekboek voor de Amiga, Bruna uitg., Utrecht, ISBN 90-229-3470-5

° Kelly, A. & I.Pohl, De programmeertaal C, grondbeginselen en toepassingen, Addison-Westley Publ.Comp. Inc., Amsterdam, ISBN 90-6789-068-5

Voorbeeld.

```
#ifdef AZTEK
 #include "functions.h"
#endif
#include "exec/types.h"
#include "exec/memory.h"
#include
"intuition/intuitionbase.h"

#define LOPRI_DOSERROR 5

struct IntuitionBase
*IntuitionBase;
struct Gfxbase *Gfxbase;
struct NewScreen g_NS = {
  0,0,320,255,4,0x00,0x01,
  NULL,CUSTOMSCREEN,
  NULL,(UBYTE *)"Commodore
    INFO  c-demo",
  NULL,NULL };
struct NewWindow g_NW = {
  0,0,320,255,0x00,0x01,NULL,

    SMART_REFRESH|WINDOWDRAG|W
    INDOWDEPTH,
  NULL,NULL,(UBYTE
    *),"COMMODORE INFO
    C-DEMO",

    NULL,NULL,0,0,0,0,CUSTOMSC
    REEN };
struct Image g_DemoImage = {
  0,0,64,16,4,
  NULL,0x0f,0,
  NULL };

UWORD g_Data[] =
{
 0x0000,0x0000,0x0000,
    0x0000, 0x0000,0x0000,
    c0x0000,0x0000,
0x0000,0x0000,0x0000,
    0x0000, 0x03c0,0x01de,
    0x77e3, 0xc000,
0x07c0,0x41de,0x77e7,
    0xe000,0x07c0,0xc1df,
    0x770e,0x7000,
0x0fc1,0xc1df, 0x770e,
    0x7000, 0x0c3f,0xc1df,
    0xf7cf,0xf000,
0x0001, 0x81dd, 0xf7cf,
    0xf000, 0x4000,0x01dd,
    0xf70e,0x7000,
0x4000,0x01de, 0xf707,
    0xe000, 0x6000,0x01de,
    0xf7c3,0xc000,
0x703f,0xc001,0x00c8,
    0x1000,0x3c3f,0xc001,
    0x0004,0x3000,
0x3fff,0xc1dc,0xf703,
    0xe000,0xffff,0xffff,
    0xffff,0xff00,

0x03c0,0x0000,0x0000,
    0x0000,0x0fc0,0x0000,
    0x0000,0x0000,
0x1fc0,0x0000,0x0000,
    0x0000,0x3fff,0xc000,
    0x0000,0x0000,
0x3fff,0xc000,0x0000,
    0x0000,0x7c3f,0xc000,
    0x0000,0x0000,
0x783f,0x8000,0x0000,
    0x0000, 0x783f,0x0000,
    0x0000, 0x0000,
0x783e,0x0000,0x0000,
    0x0000, 0x783f,0x0000,
    0x0000, 0x0000,
0x7c3f,0x8000,0x0000,
    0x0000,0x3fff,0xc000,
    0x0000,0x0000,
0x3fff,0xc000,0x0000,
    0x0000,0x1fff,0xc000,
    0x0000,0x0000,
0x0fc0,0x0000,0x0000,
    0x0000,0x03c0,0x0000,
    0x0000,0x0000,
0x03c0,0x0000,0x0000,
    0x0000,0xffff,0xffff,
    0xffff,0xff00,
0xffff,0xffff,0xffff,
    0xff00,0xffc0,0x3e21,
    0x881c,0x3f00,
0xffc0,0x3e21,0x8818,
    0x1f00,0xffc0,0x3e20,
    0x8810,0x0f00,
0xffc0,0x7e20,0x8810,
    0x0f00,0xffc0,0xfe20,
    0x0831,0x8f00,
0xffff,0xfe20,0x0831,
    0x8f00,0xffff,0xfe20,
    0x0831,0x8f00,
0xffff,0xfe22,0x0830,
    0x0f00,0xffff,0xfe22,
    0x08f0,0x0f00,
0xffff,0xfe23,0x08f8,
    0x1f00,0xffff,0xfe23,
    0x08fc,0x3f00,
0xffff,0xffff,0xffff,
    0xff00,0x03c0,0x0000,
    0x0000,0x0000,

0x0000,0x0000,0x0000,
    0x0000,0x0000,0x0000,
    0x0000,0x0000,
0x0000,0x0000,0x0000,
    0x0000,0x0000,0x0000,
    0x0000,0x0000,
0x0000,0x0000,0x0000,
    0x0000,0x0000,0x0000,
    0x00e1,0x8000,
0x0000,0x0000,0x00e1,
    0x8000,0x0000,0x0000,
    0x0000,0x0000,
0x0000,0x0002,0x0000,
    0x0000,0x0000,0x0002,
    0x00c0,0x0000,
0x0000,0x0001,0x00c8,
    0x1000,0x0000,0x0001,
    0x000c,0x3000,
0x0000,0x01dc,0xf707,
    0xe000,0x0000,0x01dc,
    0xf703,0xc000,
```

```
 0x0000,0x0000,0x0000,
    0x0000,0xfc3f,0xffff,
    0xffff,0xff00 };
UWORD g_Colors[] =
{
 0x0000,0x0fff,0x0f00,
    0x0a00,0x00f0,0x00a0,
    0x000f,0x000a,
 0x0999,0x0070,0x00ff,
    0x00aa,0x0f0f,0x0a0a,
    0x0fa0,0x0a50
};
void GP_BailOut(f_How,f_Why)
int f_How;
char *f_Why;
{
 printf("%s\n",f_Why);
 exit(f_How);
}
void
    CD_TekenVoorbeeld(f_DemoScr,
    f_DemoData)
struct Screen *f_DemoScr;
UWORD *f_DemoData;
{
 struct RastPort *DRastPort;

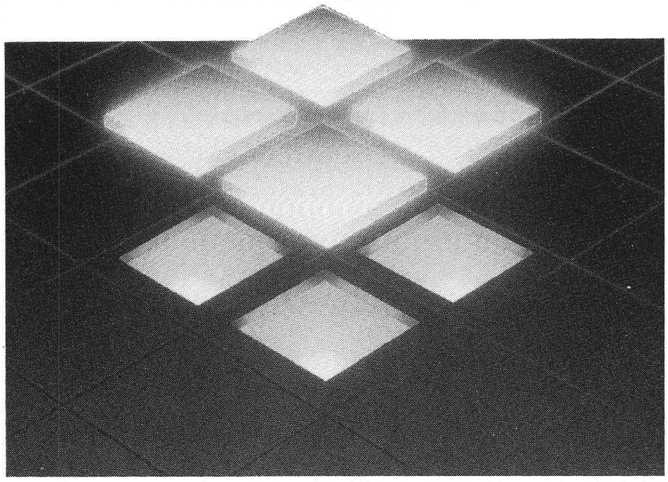
 DRastPort=&f_DemoScr-
    RastPort;
 g_DemoImage.ImageData=
    (USHORT *)f_DemoData;
 DrawImage(DRastPort,&g_
    DemoImage,100,100);
}
void CD_OpenAlles()
{
 IntuitionBase=(struct
    IntuitionBase *)
  OpenLibrary("intuition.
    library", 0L);
 if(!(GfxBase=(struct GfxBase
    *)
  OpenLibrary("graphic.
    library", 0L);
GP_BailOut(LOPRI_DOSERROR,
    "Geen graphics.library");
}
void
    CD_SluitAlles(f_DemoScreen,f
    _DemoWindow)
struct Screen *f_DemoScreen;
struct Window *g_DemoWindow;
{
 CloseWindow(f_DemoWindow);
 CloseScreen(g_DemoScreen);
 CloseLibrary(GfxBase);
 CloseLibrary(IntuitionBase);
}
void CD_WaitForCounter()
{
 int i;
 for(i=0;i0;i++)
}
main()
{
 struct Screen *l_DemoScr;
 struct Window *l_DemoWindow;
 UWORD *l_DemoData;
 register int i;
 CD_OpenAlles();
 if(!(l_DemoScr = (struct
    Screen *)
    OpenScreen(&g_NS)))
GP_BailOut(LOPRI_DOSERROR,
    "Scherm wil niet openen");
 g_NW.Screen=l_DemoScr;

 if(!(l_DemoWindow = (struct
    Window *)
    OpenWindow(&g_NW)))
GP_BailOut(LOPRI_DOSERROR,
    "Window wil niet openen");
 LoadRGB4(&l_DemoScr-
    ViewPort,g_Colors,16);
 RemakeDisplay();
 if(!(l_DemoData=(UWORD *)
    AllocMem(64*16*sizeoff(UWORD
    ),MEMF_CHIP|MEMF_CLEAR)))
  GP_BailOut(LOPRI_
    DOSERROR,"Niet genoeg
    geheugen voor image");
 for(i=0;i(16*64);i++)
  *(l_DemoData+i)=g_Date[i];
 CD_TekenVoorbeeld(
    l_DemoScr,l_DemoData);
 CD_WaitForCounter();
 FreeMem(l_DemoData,
    64*16*sizeoff(UWORD));
 CD_SluitAlles(l_DemoScr ,
    l_DemoWindow);
}
```

IN DE BAN VAN C

Lattice C v5.04

De voorgaande versie van de Lattice C compiler is nauwelijks doorgedrongen tot de Nederlandse markt of er is al weer nieuws van het Lattice front. De nieuwste Lattice C compiler, versie 5.04 inmiddels, is complexer maar ook gebruikersvriendelijker geworden. Hier volgt een test van de nieuwste telg in de Lattice C compiler familie.



Met versie 4.0 was Lattice al een eind op weg naar de 'prettig mee te werken compiler'. Het geheel functioneerde prima doch kende een aantal nadelen. Eén hiervan was het gebrek aan een geïntegreerde omgeving. In versie 5.04 wordt hieraan vrijwel geheel tegemoet gekomen. Wat zijn de belangrijkste veranderingen in het pakket.

Nieuwe eigenschappen

Zoals al duidelijk werd, kent de versie 5.04 een (bijna) geïntegreerde werkomgeving. Bijna, want vaak moet men onderdelen vanuit DOS oproepen. Een omgeving zoals men vanuit Turbo C kent, komt hier (nog) niet voor. De Lattice Screen Editor, afgekort L.S.E., maakt een belangrijk onderdeel uit van de werkomgeving. Vanuit deze editor is het mogelijk de eenmaal opgemaakte files te compilen en te linken. Wanneer LSE vervolgens verlaten wordt, is het programma klaar om uitgevoerd te worden.

Bij de test van versie 4.0 werd al snel duidelijk dat de compiler snel en gemakkelijk te gebruiken was het geproduceerde resultaat was over het algemeen flink onder de maat. Ook hier heeft men in de huidige versie rekening mee gehouden. Het pakket produceert een compactere en snellere code. Als dit naar uw zin nog niet genoeg resultaat oplevert, dan kunt u de 'Global Optimizer' inschakelen. Met dit onderdeel wordt de code geoptimaliseerd, dat wil zeggen de source file wordt zodanig gecompiled dat er een snellere en compactere uitvoerbare file ontstaat dan er geweest zou zijn zonder het optimaliseringsproces.

Wie wel eens met afgunst naar de Aztek C bezitters heeft gekeken om het gemak van hun 'Source Level Debugger' hoeft dit nu niet meer te doen. Versie 5.04 kent ook een dergelijke debugger. Lattice noemt het 'CodePRobe (CPR)'. Met dit programma wordt het mogelijk C-teksten op 'source-niveau', dus vanuit de oorspronkelijke brontekst, te controleren op fouten.

Versie 5.04 voldoet nu (bijna) aan de ANSI standaard. De compiler kent wel de volledige ANSI preprocessor.

De compiler kent een aantal nieuwe keywords. Signed dient om een default 'unsigned' optie te vervangen. Men heeft twee keywords toegevoegd om tot betere adressering van data en functies te komen. Het gaat hier om de keywords near, (relatieve adressering ten opzichte van een startadres), en far waarmee absolute adressen gegenereerd worden. Als bepaalde data zondermeer in het 'chip-ram' moet worden geplaatst, dan is het gebruik van het keyword chip op z'n plaats. Een zeer interresant mogelijkheid van de nieuwe compiler is het vermogen om door middel van keywords te bepalen of de parameters van functies via registers of via de stack moeten worden overgedragen. Samenwerking van C-programma's met assembly wordt hierdoor flink vereenvoudigd.

Om het compilersysteem ook op kleinere systemen te kunnen laten draaien, heeft men gekozen voor twee verschillende compilers. De eerste 'pass' kent twee verschillende versies, te weten lc1 en lc1b. Bij een 'ruimer' uitgeruste Amiga is er de mogelijkheid om de 'big-lc1' aan te roepen. Deze compiler kent de 'full-listing ability' waarmee men bedoelt dat het mogelijk wordt om een complete list te krijgen van uw source files. Hierbij worden de include-files volledig getoond.

Ter ondersteuning van de Shell Resident mogelijkheid genereert de compiler nu resident-compatible code. Het resident

Resident maken

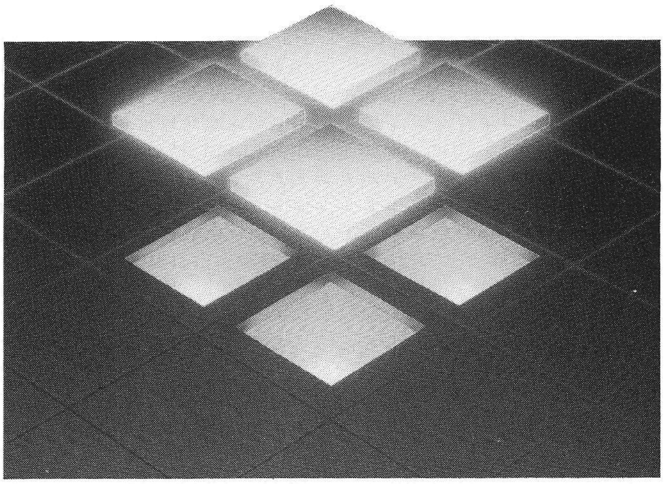
Lattice gemaakte programma's resident maken behoort nu ook tot de mogelijkheden. Tevens zijn de compiler passes lc1 en lc2 en de linker blink resident te maken. Dit verhoogt vervolgens weer de compile- en linksnelheid.

De developers uitvoering van het pakket, welke wij ter recentie aangeboden kregen, bestaat uit twee redelijk stevig lijkende ringbanden, vijf schijven en een aantal losse papieren. Voorgaande ervaringen met ringbanden deed ons een beetje sceptisch zijn, echter het geheel heeft onze

duurbaarheidstest doorstaan dus een dikke plus. De handleiding is, door het gebruik van tabs, in duidelijk te onderscheiden onderdelen verdeeld. Elk onderdeel kent vervolgens weer een aantal hoofdstukken waarin men de specifieke onderwerpen probeert uit te leggen. Ook hier weer een dikke plus. Over het algemeen zijn de hoofdstukken, ondanks dat het geheel in het Engels geschreven is, goed leesbaar. Maar we hebben nog enige aan- of opmerkingen. Sommige onderdelen, zoals compileropties of keywords, werden niet goed genoeg uitgelegd of men veronderstelde dat de achtergrond bekend was bij de lezer. Toch mogen we ook aan de leesbaarheid een ruime voldoende toekennen.
Concluderend over de handleiding mogen we stellen dat het geheel er prima en professioneel verzorgd uitziet. In het bijzonder dankzij de tabs en de in ruime mate aanwezig zijnde indexen. Achter elk hoofdstuk is er namelijk één aanwezig en aan het eind van de twee delen heeft men een zogenaamde 'master index' toegevoegd, dat .

LSE

Wie éénmaal het gemak van de werkomgeving van Turbo C van Borland heeft geproefd, wil eigenlijk niet meer terug. Of het moet niet anders kunnen. De omschakeling naar het 'met de hand' compilen op de Amiga was ondergetekenden dan ook vaak een zeer onaangename gewenning. Het was daarom dan ook een zeer prettige verrassing te merken dat Lattice met de versie 5.04 enigzins aan onze wensen tegemoet gekomen was.
De Lattice Screen Editor, afgekort LSE, bevat bijna alles wat een doorgewinterde Amiga C-programmeur zich wensen kan. Met gemak meerdere files tegelijk editten, files precompilen, optimalizeren en linken, al



dat kan zonder veel moeite gerealiseerd worden in LSE. Door in DOS 'lse [filenaam]' in te typen verschijnt na enige tijd het LSE venster op het scherm. Het is nu vervolgens mogelijk een file te bewerken. Dit bewerken kan gebeuren door de functies te selecteren uit de 'pull-down menus' of door specifieke toetsencombinaties te gebruiken. Na de editsessie kan de file meteen vanuit LSE gecompiled, gelinked en zelfs geoptimaliseerd worden. Door nu een nieuw CLI venster te openen kan, zonder LSE verlaten te hebben, het programma opgestart worden.
Naast de hierboven genoemde eigenschappen bezit LSE nog over een aantal andere handigheden, zoals automatische filebackup, het automatisch inspringen bij een openingsaccolades ({) en het weer terugspringen bij gebruik van de sluitaccolade. (})

Optimaliseren van code

De Lattice compiler staat bekend om het feit dat deze wat grotere en tragere code genereert dan dat van concurrent Aztek. In de strijd tegen dit feit heeft Lattice nu de Global Optimizer geworpen. De optimizer leest de door LC1 geproduceerde 'quad file'. De optimizer probeert vervolgens de gelezen code te verkleinen en te versnellen aan de hand van een aantal vaste optimaliserings stelregels.
Niet alleen de Global optimizer zorgt ervoor dat versie 5.04 voor een snellere code zorgt, ook een aantal nieuwe keywords resulteren in een aanzienlijke snelheidstoename. Door het op de juiste plaats gebruiken van keywords als 'near', 'far', 'huge', 'signed', '__regargs' en '__stdargs' kan men aanzienlijke snelheidswinsten ten opzichte van de 4.0 compiler boeken. De eerste drie keywords, near, far en huge beïnvloeden de adresseringswijze van data of functies. Door gebruik van het near-keyword wordt het adres van een variabele als een 16 bits offset ten opzichte van een basepointer opgevat. Het adres wordt dus in 16 bits geplaatst wat bij veel data tot aanzienlijke geheugenbesparing kan leiden. Het huge-keyword is gelijk aan het far-keyword. Dit keyword zorgt ervoor dat het adres van data of functies in 32 bits opgeslagen wordt. Het huge-keyword is alleen opgenomen om compatibiliteit met de MS-DOS versie van de Lattice compiler te garanderen.
Keywords als '__regargs' en '__stdargs' bepalen hoe de parameters van functies overgedragen worden. Normaal worden in C-programma's de parameters via de stack overgedragen. Dit in tegenstelling tot assembly routines waarbij men veelvuldig van data- en adresregisters gebruik maakt. Versie 5.04 beschikt over de mogelijkheid om het op beide manieren te doen. Wil men de parameters via de stack overdragen, dan moet het keyword '__stdargs' zoals hieronder geïllustreerd is gebruikt worden.

```
int __stdargs Test(int a, char *b)
```

Wil men echter de variabelen via registers overdragen, gebruik dan het keyword __regargs.

```
long __regargs Demo(a,c)
```

Het __asm keyword illustreert dat C en Assembly steeds meer verweven raken. Met het __asm keyword kan opgegeven worden via welke registers de parameters overgedragen worden. Wilt u precies bepalen welke registers gebruikt worden, zoals geillustreerd is in het volgende voorbeeld, dan zal het __regargs keyword niet voldoende de uitkomst bieden.

```
int __asm Som(getal1, getal2)
register __d0 int getal1;
register __d1 int getal2;
```

In dit voorbeeld worden de parameters 'getal1' en 'getal2' via registers d0 en d1 overgedragen. Met het _asm keyword in combinatie met de register keywords, kan men precies bepalen via welke registers de parameters moeten worden overgedragen.
Wil men op andere wijze de parameteroverdracht beinvloeden dan dient men het #pragma-statement te hanteren. Met #pragma's beinvloedt men de parameteroverdracht van libraryfuncties. Door aan te geven welke functie in welke library vanaf welke offset te vinden is kan men van deze functie de te gebruiken registers bepalen. Dit geschiedt door in een van

tevoren vastgesteld formaat de codenummers van de te gebruiken registers te rangschikken.
Dit is voor de Lattice bezitters oud nieuws, immers versie 4.0 bezat deze mogelijkheid ook al, echter dat de pragma's grote flexibiliteit bieden blijkt wel uit het feit dat ook in de nieuwe Aztek compiler de pragma's geintroduceerd zijn.

CodePRobe

Zoals al eerder naar voren is gekomen, is CodePRobe (CPR) de Source-Level Debugger bij Lattice. Dat deze CPR bijzonder gemakkelijk kan zijn hebben onderstaanden aan den lijve ervaringen bij het schrijven van enkele utilities. Met CodePRobe kunnen programma's op sourcebasis, dus op brontekst-basis, van fouten worden ontdaan. U kunt de inhoud van elke variabele, array of structure bekijken, programma's kunnen stap voor stap, regel voor regel doorlopen worden. CodePRobe werkt op analoge wijze als LSE. Het kan worden bedient door middel van commando's geselecteerd in de pull-down menu's of door een specifiek ingetoetst commando. Na ook dit programma vanuit DOS opgestart te hebben verschijnen er twee vensters op het scherm. Het programma zal enige data laden waarna de te debuggen source file in het bovenste venster verschijnt. Deze file kan vervolgens op drie niveaus worden aanschouwd. Men kan er op C niveau mee werken, of men bekijkt het op Assembly niveau, of, voor diegenen die er behoefte aan hebben, beide modes kunnen tegelijk worden gebruikt. Men kan vervolgens 'single stepped' door het programma heenlopen of men slaat bepaalde niet interessante gedeelten over. (proceed) Door nu watches en watchbreaks voor variabelen te gebruiken kan men de waarde van een variabele continu volgen. De tijden van variabelen printen in het DOS scherm zijn hierdoor voorgoed voorbij!
Om CodePRobe te gebruiken dient men de file met behulp van een speciale optie te compilen. Het programma wordt hierdoor wel groter, immers alle informatie voor de debugger moet worden opgeslagen, dus gebruik deze versie niet als uiteindelijke versie. Maar zonder de debug informatie zal de debugger niets kunnen debuggen.
Het gebruik van een Source Level Debugger of CodePRobe is een sensatie op zich welke men eerst geproeft moet hebben voor men er over kan oordelen. Programmeren in C wordt door dit soort programma's een stuk vriendelijker en het gedrag wordt, voor zover men bij programma's over gedrag mag spreken, een stuk aanschouwelijker.

Utilities

De uiteindelijke vriendelijkheid en bruikbaarheid van een ontwikkelpakket wordt vaak bepaald door het aard en aantal van de bijgeleverde utilities. Ook Lattice heeft voor een ruim aantal hulpprogramma's gezorgd.
Extract en Build zijn utilities voor het opbouwen van batchfiles. Hiervoor zou ook een editor gebruikt kunnen worden, echter in het geval van een rij van gelijksoortige operaties als bijvoorbeeld 'copy file1 to dest:file1, copy file2 to dest:file2 ...' komen de commando's Extract en Build uitermate van pas.
Met CXRef kan op vrij eenvoudige wijze net opgemaakte programmalistings op het outputdevice worden verkregen. CXRef kan listings met regelnummers en cross-reference reports afbeelden. Doormiddel van opties wordt ingesteld wat afgebeeld moet worden.
Diff wordt gebruikt wanneer de inhoud van twee files moet worden vergeleken. Het commando vergelijkt twee files en geeft aan waar de files van elkaar afwijken. Wanneer gebruik gemaakt wordt van de 'redirection' mogelijkheid van DOS kan men ook nieuwe files creëren.
Files is een bijzonder krachtig hulpprogramma. Het is in staat om files en of directories op te sporen, te kopiëren of te verwijderen. Zoals men in de handleiding al zegt, het heeft de eigenschappen van de drie UNIX commando's, find, cpio en rm in zich.
Global Regular Expression search and Print, afgekort GREP, zoekt in opgegeven files naar een bepaalde zoekstring. Het programma ondersteund ook wildcards en vrij uniek, u kunt bepaalde character classes opgeven. Bijvoorbeeld u zoekt naar het woord 'waterval" in de file vakantie.txt maar u bent er niet zeker van of waterval wel of niet met een hoofdletter geschreven is. U kunt GREP dan vertellen dat hij voor die letter zowel een hoofdletter als een kleine letter moet accepteren.
LMK is in velerlij opzichten gelijk aan de Make utility van UNIX en Aztek C. Met dit programma kunt u op veel eenvoudigere wijze files compilen en linken. Volgens een bepaald scriptfile, projectfile, bepaalt het programma welke files afhankelijk van elkaar zijn en zal vervolgens, volgens deze afhankelijkheidsregels, de files compilen en linken om tot een uiteindelijke uitvoerbare file te komen.
Dat met LMK meer gedaan kan worden dan op het eerste gezicht lijkt mag blijken uit het feit dat men twee hoofdstukken heeft gewijdt aan dit onderwerp. In het eerste komen de algemene zaken van LMK aan het licht, in hoofdstuk twee worden de meer gevorderde aspecten van LMK behandeld. Een kleine greep uit de mogelijkheden? LMK ondersteund 'multiple and fake targets, command line options, local input files and transformation rules'.
Met de Lattice Profiler kunt u bedoordelen of uw program-

Tabel1. Priemgetallentest

	grootte	tijd1	tijd2
Aztek v3.	6683	2,838	277,644
Latt. kaal	11688	3,295	331,562
Latt.+optimizer	11652	3,204	321,126
Latt.+optimizer en near optie	11652	3,204	331,034

Tabel2. Procentuele afwijkingen.

	grootte	tijd1	tijd2
Aztek v3.6	100	100	100
Latt. kaal	171,1	116,1	119,4
Latt.+optimizer	170,6	112,9	115,7
Latt.+optimizer en near optie	170,6	112,9	119,2

Figuur 3. Fractal snelheidstest

	grootte	tijd
Aztek v3.6	7972	10m 40,636s
Latt. kaal	14768	32m 55,455s
Latt.+optimizer	14724	32m 52,983s
Latt.+optimizer en near optie	14724	32m 53,166s

Figuur 4. Relatieve waarden tov. Aztek

	grootte	tijd
Aztek v3.6	100	100
Latt. kaal	185,2	308,4
Latt.+optimizer	184,7	307,9
Latt.+optimizer en near optie	184,7	308,0

ma efficiënt genoeg uitgevoerd wordt. De profiler bestaat uit twee onderdelen. Het omderdeel lprof meet de tijd van elk uitgevoerde onderdeel en lstat geeft de uiteindelijke resultaten in een verslag weer. Aan de hand van deze gegevens kunnen veranderingen in programma gemaakt worden, delen die niet snel genoeg uitgevoerd worden kunnen in assembly omgezet worden, etcetera.

De term SPLAT doet niet meteen denken aan een computerprogramma. De werkelijkheid is anders. Bij het Lattice C compilersysteem is SPLAT een line-editor. Het vertoont gelijkenissen met de AmigaDOS editor Edit. Toch is SPLAT niet alleen maar een line-editor. Het heeft een aantal mogelijkheden welke het krachtiger maakt dan welke andere line-editor. Het beschikt over de mogelijkheid woorden in meerdere files te vervangen. De oorspronkelijke file wordt hierbij ongemoeid gelaten, tenzij men dit expliciet wenst.

Tot slot de Touch and WordCount utilities. Met Touch kunt u voor één of meerdere files de tijd- en datumvermelding aanpassen. WordCount telt het aantal characters, woorden en regels in een file. Let er hierbij op dat WordCount een woord definieert als, een verzameling van characters afgesloten door een spatie. WordCount zou in het voorbeeld 'I would die 4 U' 5 woorden geteld hebben.

Snelheid en omvang

Om de prestaties van de nieuwe compiler een beetje beter in beeld te krijgen hebben we een aantal tests uitgevoerd. We hebben de Lattice 5.04 vergeleken met de 'oude' Aztek v3.6 compiler, omdat de Aztek v5.0 compiler op het tijdstip van schrijven nog niet in ons bezit was. Tabel 1 geeft de resultaten van de priemgetallen test weer. Met de Lattice 5.04 compiler is de test op drie verschillende wijzen uitgevoerd. De eerste test is zonder enige opties, keywords en zonder profiler uitgevoerd. In het tweede geval is de profiler er aan de pas gekomen. Bij de derde uitvoering zijn zowel de profiler als de -b1 optie toegepast. Met de -b1 optie worden alle variabelen als 'near' opgevat.

Uit de test (tabel 1)blijkt dat de grootte van de door de Lattice 5.04 compiler geproduceerde files nog steeds groter zijn dan het resultaat van de Aztek compiler. Zelfs na gebruik van de optimizer of near optie kan het resultaat nog niet in de buurt komen van de Aztek files.

De tijdresultaten geven een wat genuanceerder beeld. We hebben twee runs uitgevoerd met dezelfde priemgetallen test. 'tijd1' geeft de tijd weer die het programma nodig had om uit de eerste 100 getallen de priemgetallen te vinden. Bij 'tijd2' wordt het tijdresultaat na 1000 getallen weergegeven. Nog even voor alle duidelijkheid, de tijdseenheden zijn seconden.

Om van deze absolute cijfers een duidelijker beeld te krijgen hebben we een tweede tabel opgezet. Deze tabel bevat de relatieve verschillen ten opzichte van de Aztek v3.6 compiler.

Uit deze cijfers (tabel 2) blijkt dat, in het bijzonder met gebruik van de optimizer, dat de tijdresultaten de Aztek compiler niet veel ontwijken.

Om een eenzijdig beeld te voorkomen hebben we een tweede snelheidstest gebruikt. Het gaat hierbij om een programma dat fractalfiguren creert. Gemeten is wederom de grootte van het programma en snelheid van uitvoering. Er dient nog wel even opgemerkt te worden dat de tijdmeting begint vanaf het begin van de test tot het einde, waarbij de taskswitching gedurende de test uitgeschakeld wordt.

Ook van figuur 3 hebben we een relatieve versie geproduceerd. Ook hier zijn alle getallen weer ten opzichte van het door de Aztek v3.6 compiler gemaakte resultaat.

Ook uit deze tabellen blijkt dat Lattice nog steeds grotere files dan Aztek produceert. Wat bij de tweede test echt opvalt is het verschil in executiontime. De Lattice compiler doet er ruim 3x zo lang over om het fractal figuur te berekenen. Gezien de resultaten van de priemgetallentest kunnen we niet stellen dat het aan de rekenkundige vaardigheden van de compiler ligt. We moeten naar alle waarschijnlijkheid het eerder zoeken in de beeldbewerkingsfuncties. Om nu volledige reproduceerbaarheid van de test te garanderen hebben we hieronder nog een aantal gegevens over de test afgebeeld.

+ AngleR = -2.00
+ AngleI = -1.25
+ Side = 2.50

+ hoofdloop van het programma:

```
for(y=0;y<=200;y++)
{
 for(x=0;x<=200;x++)
 {

CR=(DOUBLE)x*DistanceX+A
ngleR;

CI=(DOUBLE)y*DistanceY+A
ngleI;
  ZR=CR; ZI=CI;
  Iteration=0;
  Length=0.00;
  while(Length < 3.00 &&
Iteration<33)
  {
   A=ZR*ZR;
   B=ZI*ZI;
   Length=sqrt(A+B);
   ZI=2*ZR*ZI+CI;
   ZR=A-B+CR;
   Iteration++;
  }
  DrawPixel(x,y,Iteration);
 }
}
```

Conclusies

Dankzij de uitermate gebruikersvriendelijke omgeving, de veelheid aan mogelijkheden en het goede handboek komt de compiler flink positief uit de bus. Jammer blijft echter nog steeds dat de code groter is dan dat wat concurrent Manx produceert. Vreemd blijft ook het verschil in uitvoertijd tussen de fractal- en de priemgetaltest.

Het pakket is zondermeer een aanrader voor wie zinvol programmatuur in C wil ontwikkelen. Dankzij de steeds groter worden conformiteit aan de ANSI C standaard wordt het steeds eenvoudiger om programmatuur voor meerdere besturingssystemen te ontwikkelen. Mogen wij u er nog even op wijzen dat C nog altijd een systeem programmeertaal is. Voor diegenen die het pakket aan willen schaffen moeten echter wel rekenen op een 'flinke' aanslag op de portefeuille. Maar voor de ontwikkelaar, in spé, zal dat een zorg zijn. Elke minuut ergenis met een niet goed, of niet goed genoeg werkend pakket kost immers veel meer geld. En het zou zelfs het eindresultaat negatief kunnen beinvloeden. De firma Lattice Inc. zal met de introductie van dit pakket het professionele marktsegment weer verder naar zich toe kunnen halen.

Johan & Johan

Beoordeling: 8.7 (10 max.)

Behandeld: De nieuwe Lattice compiler kent een geintegreerde werkomgeving, beschikt over een SLD en optimizer.

Positief: Zeer gebruikersvriendelijk. Dankzij CodePRobe zeer gemakkelijk fouten op te sporen. De LSE vereenvoudigd het editten, compilen en linken aanzienlijk.

Negatief: Helaas nog steeds te grote code en een tragere uitvoering van deze code.

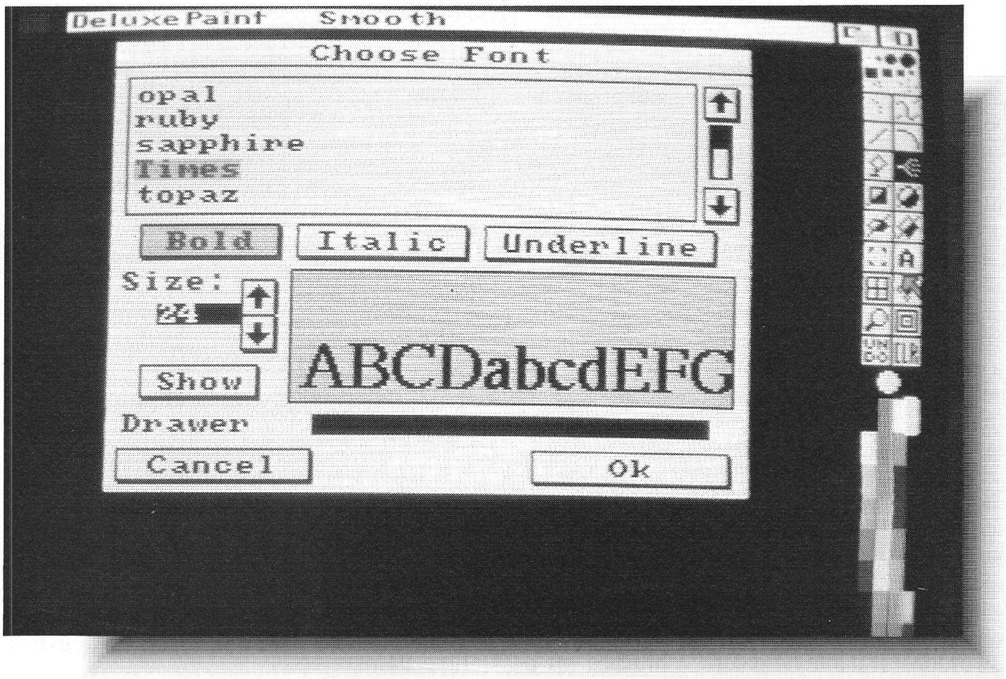
Produkt: Lattice C v5.04 compiler.
Lattice, Inc.
2500 S.Highland Avenue
Lombard, IL 60148
USA

Nederlandse leverancier: Fransen Automatisering
Postbus 221
3500 AE Utrecht
telefoon: 030-340418
Prijs: *f* 799,- (inclusief BTW)

DELUXE PAINT III

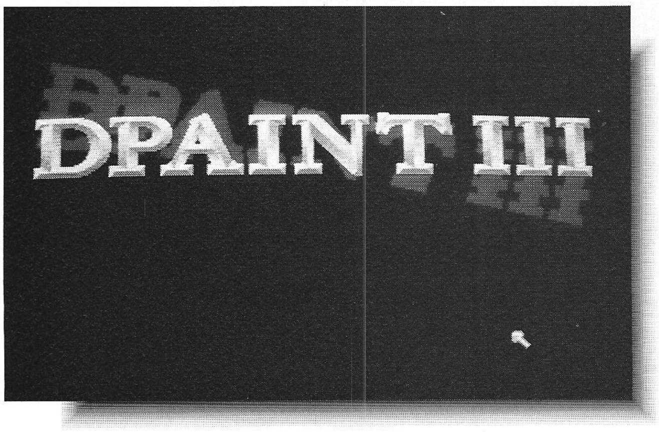
State Of The Art graphics met animatie

Versie III van Electronic Arts's Deluxe Paint III is even bij ons blijven liggen. Niettemin verdient dit grafisch zeer krachtige Amiga-pakket een bespreking. Met name de desktopvideo'ers, spelletjesmakers, grafisch artiesten en reclame-ontwerpers zullen Dan Silva's nieuwste geesteskind kunnen waarderen. Voor de echte beginner is Deluxe Paint III wellicht wat te moeilijk, maar alles valt te leren, nietwaar?



Geschiedenis

Al op de Amiga 1000 was Deluxe Video een begrip. Tal van flitsende show-effecten werden eerst op Deluxe Video getekend en later geanimeerd. Artiesten maakten er artwork, logo's en reclameposters mee. De desktopvideo- en animatiefilmmaker creëerden er hun produkties mee en ook tal van spelgraphics ontsproten aan dit Amiga-paintpakket.



Deluxe Video II was zelfs zo succesvol dat het in de PC-versie met de al overvloedig aanwezige MS-DOS-paint-software kon concurreren. Het succes op de Amiga kon op de IBM compatibele PC echter niet geëvenaard worden.

Inmiddels ligt Deluxe Paint III al enkele maanden in de schap van uw computershop. De tekenopties werden verder uitgebreid en verbeterd terwijl een aardige animatormodule werd toegevoegd. Voor rond de *f* 270,- krijgt de 500-, 1000,- 2000-, 2500- en 3000- bezitter de State Of The Art aan Amiga-graphics in huis.

Wat is nieuw?

De Amiga is door de jaren heen volop in ontwikkeling gebleven. Nieuwe videomodi, de ARexx-programmeertaal, snellere (co-)processors, geheugenuitbreidingen en vooral sterk verbeterde software maken deze Commodore-lijn nog steeds tot een moderne MC 680X0-PC. Electronic Arts volgt deze ontwikkelingen uiteraard op de voet en past haar opeenvolgende software-releases aan de recente ontwikkelingen aan.

Zo ook bij Deluxe Paint III. Iets wat release I en II niet konden was eenvoudig animeren. Om de met Deluxe Paint ontworpen graphics toch in beweging te kunnen krijgen was aanvullende animatie-software zoals bijvoorbeeld Deluxe Video of Fantavision nodig. Nu kan Deluxe Paint III ook **zelf animeren**. Verrassend eenvoudig zelfs. Via de zogenaamde **animpainting**-methode volgt de opname van de animatiese-



quenties automatisch de bewegingen van uw tekenpenseel. M.a.w.: de gebruiker tekent eerst een figuur of object, start de animatierecording en beweegt vervolgens de tekening als een penseel (Deluxe Paint-brush) door de 3D-ruimte. Later kunnen de gemaakte penseelbewegingen dan vloeiend afgespeeld en desgewenst op video of in spelletjes worden opgenomen.
De tweede belangrijke verbetering betreft de beschikbare videomodi. **Extra Halfbrite** biedt meer kleuren en twee extra fills. Met name het creëren van de schaduwen en de hoge lichten (highlights) is er een stuk eenvoudiger op geworden. Of u alle 64 paletkleuren (uit een totaal palet van 4096) tegelijk kunt gebruiken hangt van het beschikbare geheugen en de aanwezigheid van de Extra Halfbrite-chip in de Amiga af. In oude 1000-modellen blijft het palet om deze reden vaak beperkt tot 32 kleuren!
De voornaamste functie van Extra Halfbrite is het maken van graduele licht- en schaduweffecten. Ideaal voor het scheppen van diepte, interessante verlichting in artwork en voor video- animaties.
De **Overscan Painting** is een belangrijke derde verbetering. Videowerkers die zich blauw, grijs of zwart ergerden aan de lelijke randjes om het grafisch beeld komen nu volledig aan hun trekken. Mits de hardware het allemaal ondersteunt kunt u nu het gehele videoscherm vol met grafisch beeld zetten. Dit alles uiteraard volgens de PAL-norm. N.B.: Deluxe Paint III blijft tevens de mogelijkheid bieden om de NTSC-graphics (= volgens de Amerikaans TV- en videonorm) te bewerken.
Centraal in Deluxe Paint staat zoals bekend de **brush**. Een brush of "penseel" is een tekenvorm waarmee de ontwerper op het beeldschermcanvas tekent. Die vorm kan in principe van alles zijn. Een lijn, punt, airbrush-effect, geometrisch figuur of een getekend object. Daarmee is het zeer eenvoudig gecompliceerde, repeterende of 3D-beelden te creëren. U kunt de brushes uit de Deluxe Paint brush library inladen of zelf ontwerpen. In versie III werden de brushes **Tint** en **Extra Halfbrite** toegevoegd. Tint beïnvloedt de kleurverzadiging en -schakering. Met de Extra Halfbrite-brush ontstaat de beschreven lichtschaduwwerking.
Verder zijn nieuw of vernieuwd:

- De **filltypen**: Wrap (maakt het mogelijk om een patroon met behoud van horizontaal en verticaal perspectief om een 3D- object te "wikkelen"), Tint, Brush (= een penseel als fill) en Halfbrite
- **Filled** en **outlined Shapes**
- Verbeterde **printerbesturing**
- Een **filled freehand shape-tool**
- **Picture flipping** (horizontaal en verticaal zonder een brush-tussenstap zoals bij de oudere Deluxe Paint-versies
- **Automatische outline/edge**-optie
- De **stencils beschermen nu ook tegen de brushes**
- Het fontmenu is vervangen door een gebruiksvriendelijker **Font Requester**
- **Automatische grid-resizing**
- De brush kan nu **in elke hoek en vanuit elke positie gericht worden**

Als laatste nog wat kleinigheden, zoals een verbeterd manual bij de beschrijving van de 3D-effecten, de optie **No Icons** en **Auto Transparency** (= doorzichtige brush zonder het meenemen van een egale achtergrond).

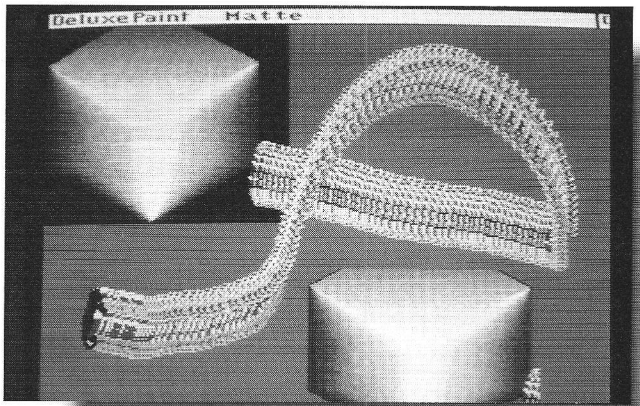
Animaties

De meest in het oog springende aanvulling bij Deluxe Paint III is de **animatiemodule**. Deze staat op een aparte disk. Animaties bestaan uit een reeks beeldjes (frames) die met een zodanige snelheid achter elkaar op het scherm worden geprojecteerd dat het menselijk oog deze beeldsequentie als één vloeiende beweging ziet. De oude tekenfilmmakers moesten daarvoor beeldje voor beeldje optekenen, overlays verschuiven en met de camera frame voor frame opnemen.



Een Amiga kan dat allemaal veel sneller. Zijn de figuren of objecten eenmaal getekend en is de bewegingsrichting uitgestippeld dan verzorgen de hardware (animatiecoprocessor) en software dat er een vloeiende animatiebeweging berekend en vervolgens ook uitgevoerd wordt.

Deluxe Paint gaat bij het ontwerpen van animaties uit van een slim doch simpel concept. Maak gewoon een reeks plaatjes die steeds net iets van elkaar verschillen en zet hen als een reeks tekenpagina's (paint-pages) in het video-RAM of op diskette. Speel (=play back) deze pages snel na elkaar af en je krijgt één vloeiende animatie.



Het gehele animatieproces komt op de volgende drie fasen neer:

- Het aanmaken van de benodigde frames
- Het tekenen op deze frames
- De **animpainting** met de brush

De beginner kan het beste een bestaand object of figuurtje pakken en alle frames via CLR en de Options Requester voor de animatie reserveren. Pak het te animeren object als brush op en druk op de linker A-toets. (of Commodore key bij de Amiga 500). Vervolgens met ingedrukte linker muisknop op het scherm de animatiebeweging(en) tekenen. De benodige frames volgen automatisch zoals de gebruiker in de oplopende teller kan zien. Alle berekeningen worden verder door de Amiga verzorgd.

Het resultaat kunt u beoordelen door Control>Play uit het Anim- menu te kiezen of gewoon de **4** aan te slaan. Met de spatiebalk wordt de lopende animatie gestopt.

Voor het animeren in 3D heeft de animator in spé de **Move Requester** nodig. Met behulp van dit animatie-tool kunt u de bewegingen op de X-, Y- en Z-as met de brush aangeven en zelfs asrotaties creëren. Desgewenst zijn ook cyclische animaties en het gebruik van een animatiereeks als brush mogelijk.

De professionele animator zal wellicht voor het **tekenen op de afzonderlijke frames** kiezen. Dit kost uiteraard veel meer tijd en inzicht. De resultaten zullen echter in de meeste gevallen vloeiender, natuurgetrouwer en veelzijdiger uitpakken.

Verder biedt Deluxe Paint III nog een scala aan animatie-effecten waarop de ware liefhebber zich helemaal kan uitleven. Bijvoorbeeld:

- Dissolves
- Expanding cirkels
- Verandering van perspectief
- Trails (spooreffecten)
- Gebogen titels
- Tumbling 3D-objecten
- Het scrollen van de achtergrond
- Stuiteren
- Geanimeerde fills

Wie het allemaal wil uitproberen is zeker een maandje zoet. Bedenk dat bij complexe animaties 1 MB aan vrij video-RAM al snel te krap blijkt. Amiga 2500/3000-bezitters zullen daar natuurlijk geen last van hebben. Bij de lagere lijn beveel ik minimaal 1MB en liefst 2MB aan.

Praktijk

Electronic Arts is er in geslaagd het trendsettende Amiga- paintpakket Deluxe Video in versie III aanmerkelijk te verbeteren en uit te breiden. Eigenlijk biedt dit pakket alles wat de grafisch artiest, spelletjesmaker, dtv-er of reclame-ontwerper nodig heeft. De animatie is een welkome aanvulling, hoewel sommige meer gespecialisdeerde videopakketten meer mogelijkheden bieden.

Het Engelse handboek is duidelijk en zal binnenkort wel door een Nederlands Deluxe Video III-boek gecompleteerd worden. Op de diskettes staan voldoende voorbeelden om direct aan de slag te kunnen. Kortom: veel waar voor uw kleine 300 piek. Blijft over de eigen creativiteit en de benodigde leertijd. Zonder eigen artistiek talent en veel experimenteren kom je met Deluxe Paint III niet verder dan wat naäpen. En dat is doodzonde van een pakket met zoveel mogelijkheden!

U.S.